Implementación de un controlador difuso en PLC

Víctor Alfonso Castañeda Arbeláez

Universidad Tecnológica de Pereira

Programa de ingeniería eléctrica

Pereira, Risaralda, Colombia

2019

Implementación de un controlador difuso en PLC

Víctor Alfonso Castañeda Arbeláez

Proyecto presentado como requisito para optar al título de:

Ingeniero Electricista

Director: Msc. Duberney Murillo Yarce

Programa de Ingeniería Eléctrica

Universidad Tecnológica de Pereira

Pereira, Risaralda, Colombia

2019

Agradecimientos

Quiero agradecer primeramente a Dios por su bondad y generosidad, porque ÉL nunca me ha desamparado y siempre ha sido mi guía. También quiero agradecer a mi familia por su apoyo incondicional y paciencia, a mis amigos y todas las personas que han estado en mi vida ya que de una u otra manera con sus enseñanzas y vivencias han forjado la persona que soy hoy en día. Quiero agradecer de manera especial a mi tía María Olga Castañeda por la confianza depositada en mi, por su apoyo incondicional y porque gracias a ella hoy puedo cumplir el sueño de ser profesional, a mi tío Manuel Castañeda por inculcarme el amor por la ciencia y el conocimiento, a Steven Sepúlveda porque aun recuerdo con total admiración sus capacidades autodidactas para aprender, Steven siempre estaré para tu hijo de manera incondicional ayudando a criarlo de la mejor manera. Pero en especial quiero agradecer a mi padre Pablo Emilio Castañeda Aguirre, porque gracias a él y sus incuestionables valores, cualidades y consejos soy una persona de bien. GRACIAS por tu apoyo, tu generosidad, tu preocupación y tu incondicionalidad. GRACIAS PAPÁ por estar siempre ahí, por tus consejos, tus enseñanzas y todo lo que has aportado a mi vida. Este triunfo también es tuyo.

Contenido

Contenido de figuras

Contenido de tablas

1. Automatización industrial

La automatización industrial es el desarrollo de un proceso de forma autónoma con una intervención mínima del trabajador. Su origen se remonta a la revolución industrial y entre los inventos más importantes de esa época se encuentran el motor de combustión externa y las máquinas textiles. Gracias a estas máquinas y la producción en serie, los trabajadores sin ser mano de obra calificada empezaron a aumentar de manera exponencial la capacidad de producción y a su vez optimizar costos y tiempos de fabricación. Años después hubo una mejora significativa en los procesos con la aparición de la lógica cableada, donde muchas técnicas que se realizaban de manera manual fueron reemplazadas por equipos eléctricos y electromecánicos tales como relés, temporizadores y contactores.

A mediados del siglo XX empezó a crecer la exigencia en los procesos industriales mediante lógica cableada y fue necesario personal altamente calificado para el diseño, montaje y mantenimiento de las instalaciones. Estas exigencias se tradujeron en pérdidas económicas cuando había daños en los elementos o había modernización de los componentes de las instalaciones, ya que el personal especializado tenía que recablear de manera individual las máquinas y los paros en la producción eran demasiado largos. La solución a esta problemática dio origen a la lógica programada, que es un diseño implementado en un circuito integrado y que permite ser reconfigurado por medio de lenguajes de programación.

Para garantizar el correcto funcionamiento de los procesos industriales es necesario el uso de controladores. En general, los controladores trabajan bajo el mismo principio: reciben una señal de entrada, la procesan y la comparan con el valor esperado ejecutando una estrategia de control. Entre los controladores más usados en la industria se encuentran los PLCs, que son aparatos electrónicos que operan de manera digital y almacenan en la memoria un algoritmo de programación para controlar en tiempo real procesos por medio de entradas y salidas digitales o analógicas.

Los PLCs están regidos por el estándar IEC 61131, de acuerdo a esta norma, los PLC deben estar en la capacidad de entender desarrollos realizados en cinco lenguajes los cuales son: Diagrama Ladder (LD), Diagrama de Bloques Funcionales (FBD), Texto Estructurado (ST), Listado de

Instrucciones (IL) y Gráfico de Funciones Secuenciales (SFC) el cual describe gráficamente el comportamiento secuencial de un programa de control.

Para el uso de estos lenguajes se necesitan software que posean un entorno de desarrollo de programación. En el siguiente capítulo se muestran los programas utilizados en este trabajo de grado.

2. Software

CoDeSys es un software de automatización donde se realizan desarrollos con lenguajes de programación pertenecientes a la norma IEC 61131-3, su nombre proviene de un acrónimo en inglés que significa sistema de desarrollo de controladores y fue creado para controlar sistemas en ingeniería.

Para iniciar un proyecto en CoDeSys V2.3 se da clic en *File>>New* y aparecerá un cuadro llamado *Target Settings* como el que se muestra en la Figura 1. En *Configuration* se observa que hay un desplegable el cual muestra el autómata al cual se desea conectar el software de programación CoDeSys; la opción *None* se usa solo para trabajar dentro del programa y ***3S CoDeSys SP PLCWinNT V2.4*** es un PLC virtual que sirve para conectarse con programas de simulación.

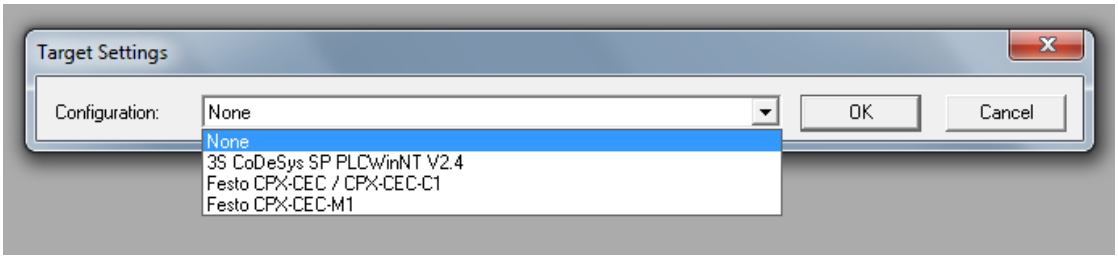

Figura 1 Target Settings

Después de hacer la elección en *Configuration* se abre una nueva ventana llamada *New Pou (Program Organization Unit)*. Allí se le asigna el nombre al POU como también se escoge su tipo, este puede ser *Program, Function Block* o *Function*, en esta última opción se puede escoger el *Return Type*, donde aparecen muchas opciones para el tipo de función que se desee, puede ser del tipo Booleano, entero, real, vector, entre otros. En la Figura 2 también se pueden observar los

lenguajes pertenecientes al estándar internacional IEC 61131-3 y el CFC, que no pertenece a la norma pero CoDeSys lo incluye como una alternativa de programación.

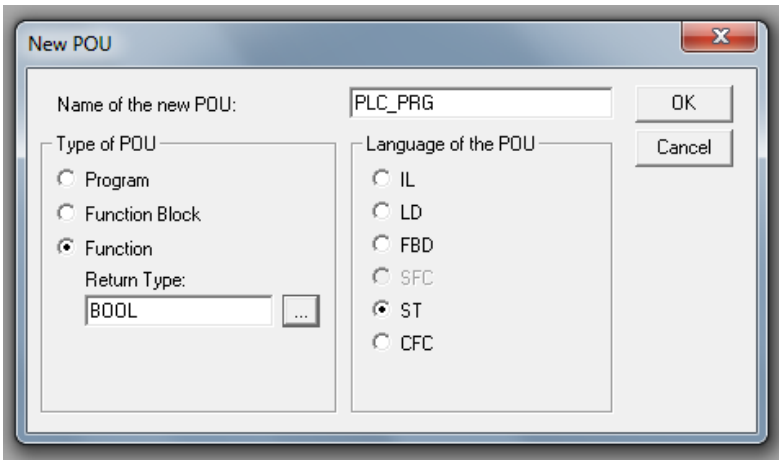

Figura 2 New POU

Cuando se da clic en OK, aparecerá la ventana de trabajo (figura 3). Allí se tiene una variedad de opciones para el desarrollo del programa.

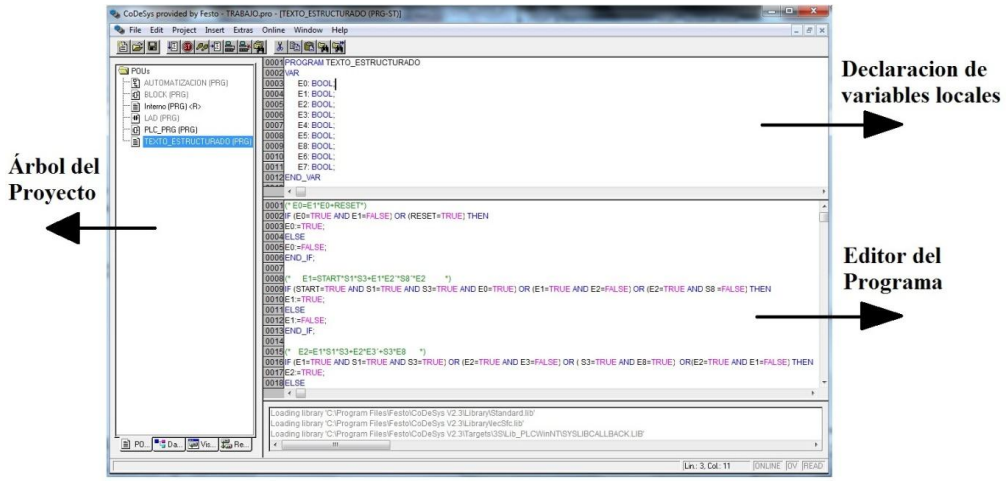

Figura 3 Ventana de trabajo en CoDeSys V2.3

Es recomendable declarar variables globales en el caso de hacer una conexión al PLC, para esto se presiona en la pestaña *Resources* que se encuentra ubicada en la parte inferior izquierda junto con otras opciones.

Entre estos botones se encuentra *Visualizations* (Figura 4), esta opción permite la creación de una interfaz gráfica que permite mostrar las etapas del automatismo y así tener certeza que no hay

errores en el código de programación. Para esto se da clic derecho debajo de la carpeta *Visualizations* y se escoge la opción *Add Object.*

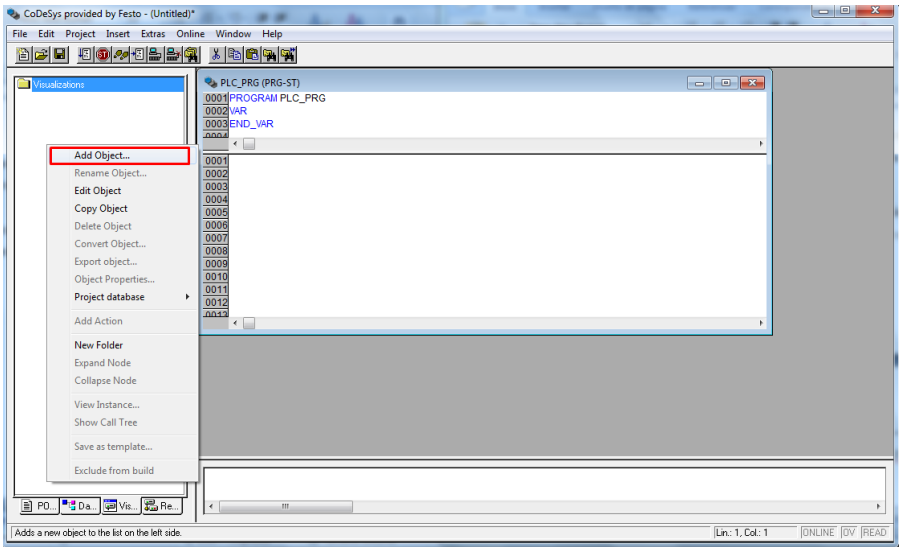

Figura 4 Adherir objetos

En la visualización se pueden insertar botones, tablas, alarmas, histogramas, sliders, Figuras geométricas y animaciones provenientes de otros programas que pertenezcan a Festo.

Dentro de esta gran variedad de opciones se pueden representar etapas por medio de figuras y que estas cambien de color cuando estén activas o inactivas, para esto se inserta la figura que se desea y se da doble clic sobre ella apareciendo una ventana como se muestra en la figura 5. Luego se señala la categoría *Colors* y se encuentran dos opciones. En *Color* se escoge para configurar el color cuando la etapa esté inactiva y *Alarm Color* cuando la etapa tenga un estado activo.

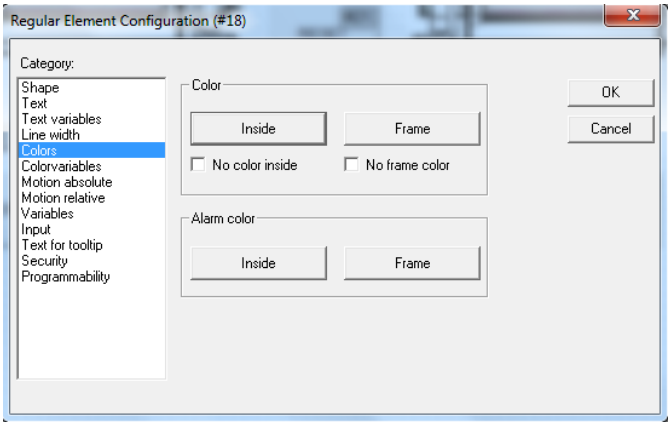

Figura 5 Configuración de elementos

Para activar el cambio de color de la figura, se hace clic en *Variables* y en *Change Color* se pone la etapa o transición donde se quiere realizar el cambio de color como se muestra en la figura 6. Para poner una de estas variables se debe tener en cuenta si se encuentra declarada en las variables globales o locales; si se encuentra en las variables locales, se presiona la tecla punto (.) seguido del nombre del POU, allí aparecerá una lista de todas las variables locales declaradas. Si por el contrario, la variable que queremos se encuentra en las variables globales, solamente se presiona la tecla punto (.) y se desplegaran todas las variables globales declaradas.

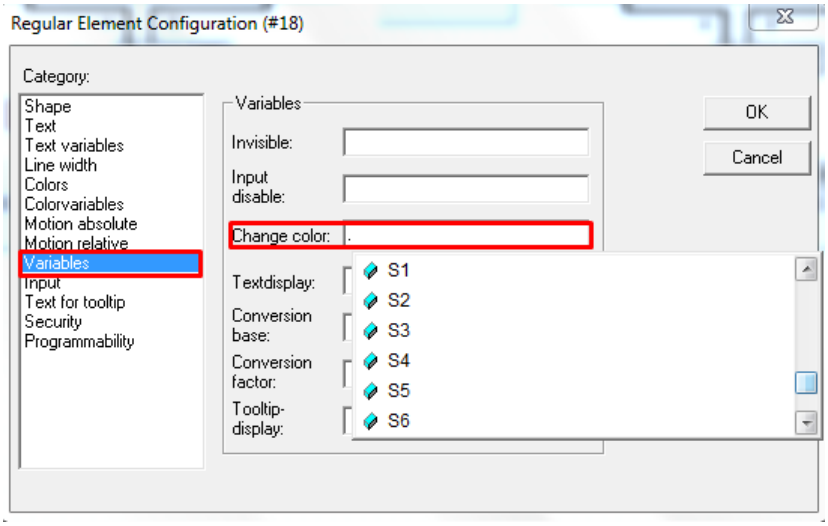

Figura 6 Cambio de color para el elemento

A continuación se muestra un ejemplo de programación en CoDeSys. En el siguiente problema se utiliza un transportador continuo para llevar una pila de paneles al punto de transferencia en el que la banda transportadora se detiene por una señal del sensor. Una vez se detiene la cinta transportadora, se activa el primer cilindro neumático el cual actúa en posición vertical donde se extiende el vástago hasta detectar que hay un panel e inmediatamente se activa el brazo extractor por succión para coger uno de ellos. Hecho esto, el vástago del cilindro se encoge hasta llegar a su posición inicial. En ese momento, actúa el segundo cilindro neumático, el cual desplaza su vástago de manera horizontal todo el mecanismo compuesto por el primer cilindro y el brazo extractor por succión junto con el panel. Cuando el mecanismo llega a la posición deseada el vástago del primer cilindro se extiende nuevamente hacia abajo hasta que el segundo sensor detecta el panel, en ese momento las ventosas sueltan el panel y se activa la segunda banda transportadora. Después de este proceso el vástago del primer cilindro se encoge y el segundo

cilindro opera de la misma forma para ejecutar de nuevo el proceso. El proceso mencionado anteriormente se muestra en la figura 7.

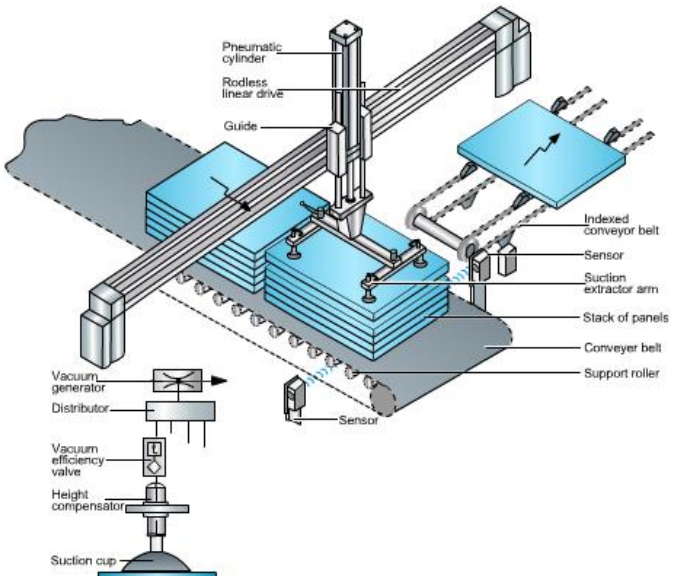

Figura 7 Sistema automatizado

A continuación se resuelve el problema secuencial mencionado anteriormente en el software de programación CoDeSys V2.3 usando los lenguajes pertenecientes a la norma IEC 61131-3. También se muestran las secuencias de operación del proceso automatizado creado en *Visualizations* (Figura 8).

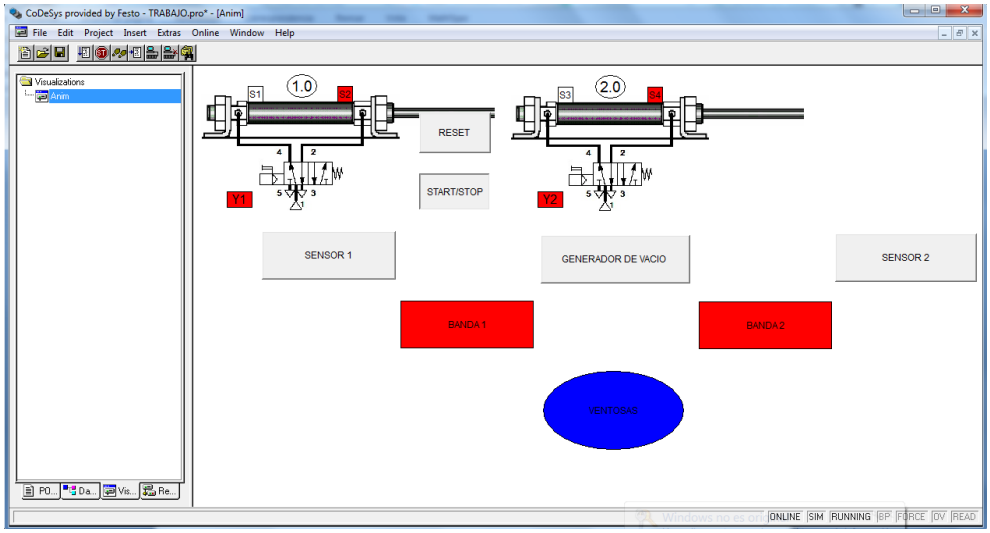

Figura 8 Secuencias de operación del proceso automatizado.

Se resuelve el problema planteado anteriormente por medio de GRAFCET ya que a partir de este modelo de representación grafica se pueden obtener ecuaciones que permiten desarrollar la lógica en todos los lenguajes de programación. Como el lenguaje de programación SFC tiene relación directa con GRAFCET se desarrolló primero el programa en dicho lenguaje (Figura 9).

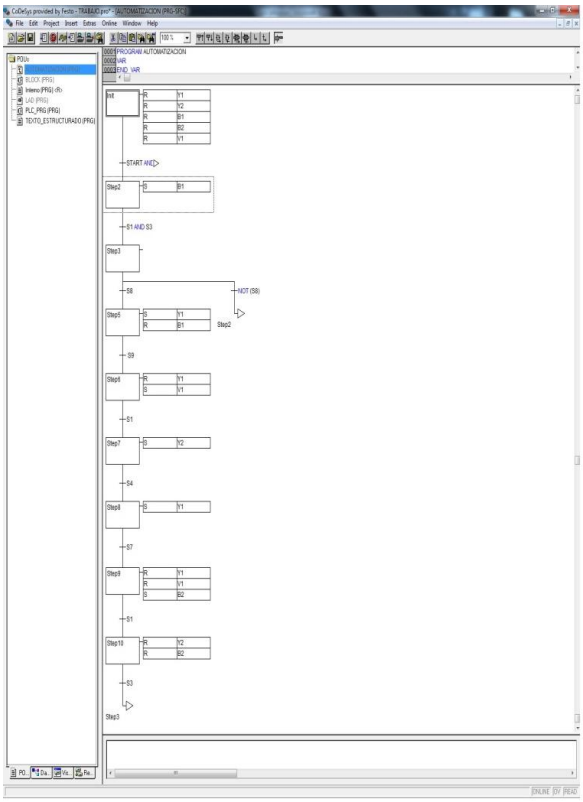

Figura 9 Programación en lengua SFC

A continuación se muestra la ecuación general del diagrama de control que permite pasar a cualquiera de los lenguajes pertenecientes al estándar IEC 61131-3 de una forma más sencilla.

$$\boldsymbol{Etapa\ Activa = Etapa\ anterior * Transicion\ anterior + Etapa\ actual * Etapa\ siguiente\ negada}$$

Ecuación 2-1

Se hace la aclaración, que las operación de suma (+) y multiplicación (*) empleadas en la ecuación 2-1 son operaciones lógicas y no aritméticas.

A continuación, en las figuras 10, 11, 12 y 13 se muestran fragmentos de los demás lenguajes de programación pertenecientes al estándar internacional IEC 61131-3.

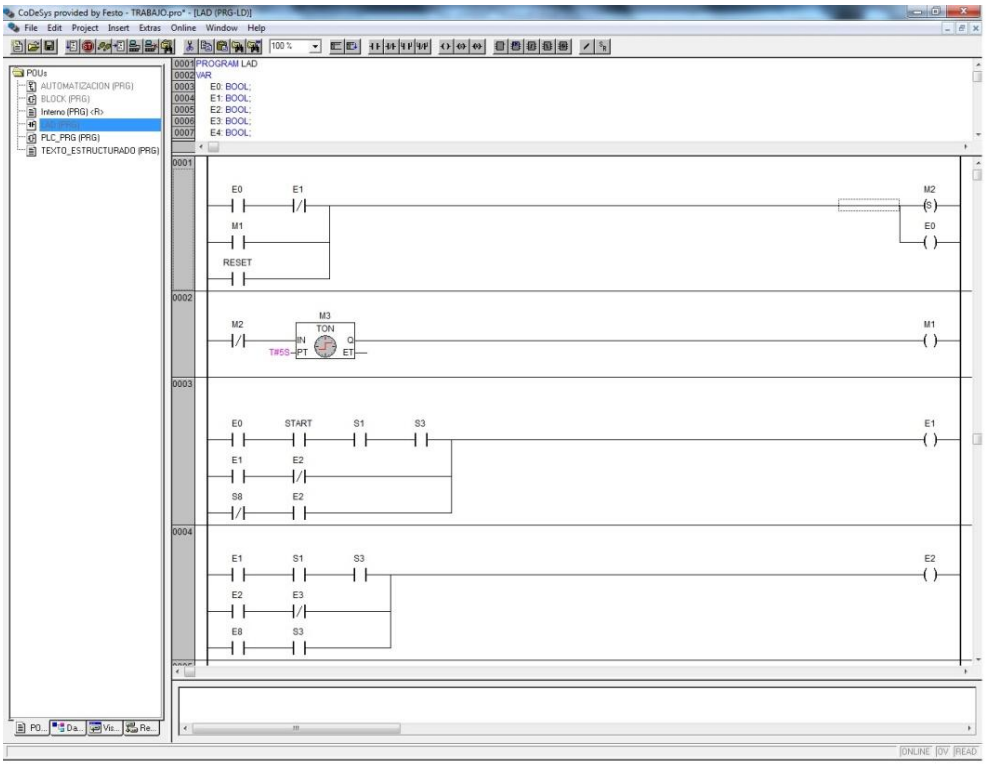

Figura 10 Programación en lenguaje escalera

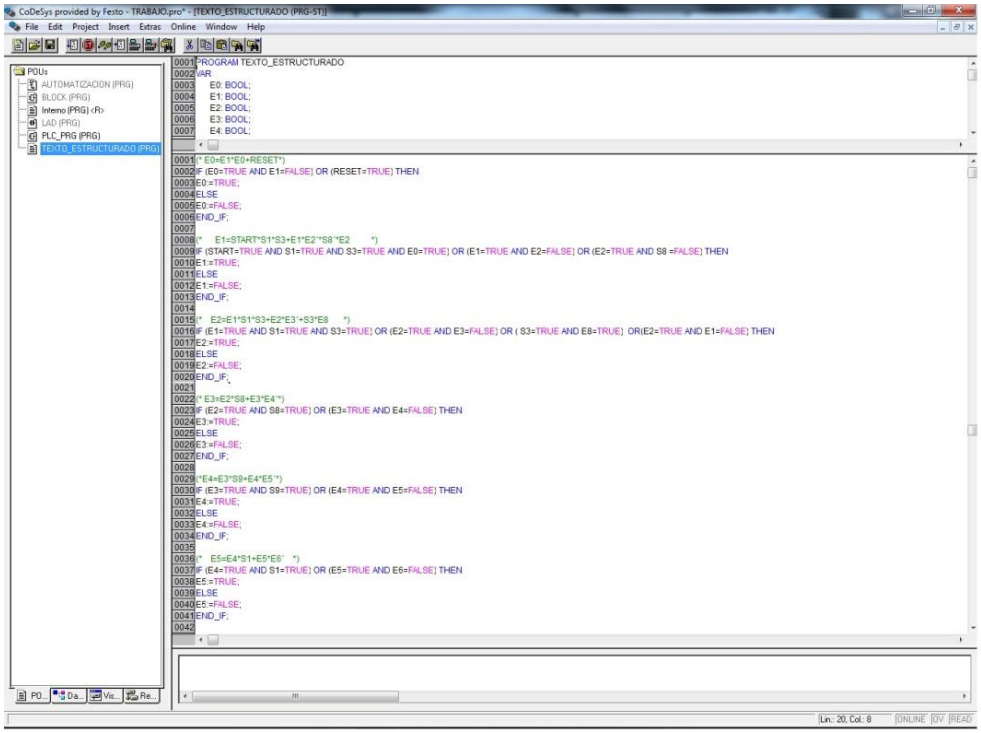

Figura 11 Programación en Texto Estructurado

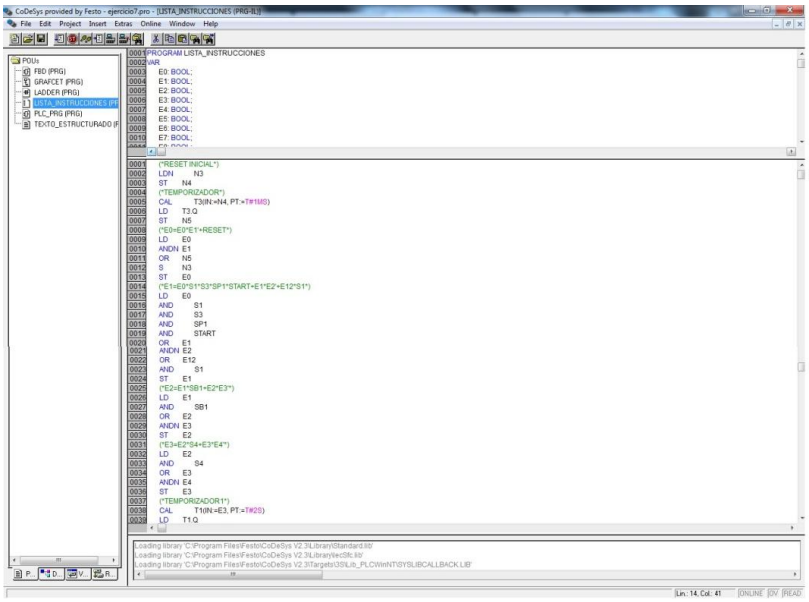

Figura 12 Programación en Listado de Instrucciones

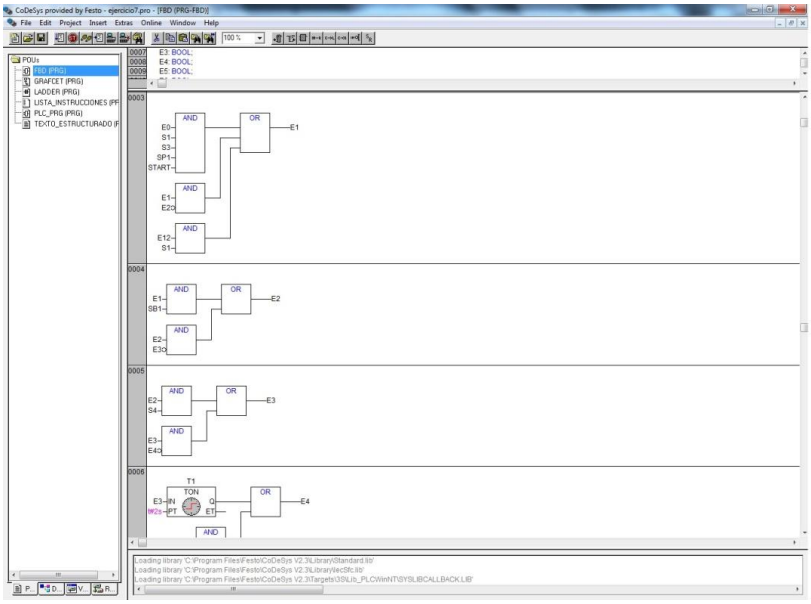

Figura 13 Programación en FBD

Un buen complemento para CoDeSys es el simulador FACTORY I/O ya que permite construir de manera rápida y eficiente escenas comunes que se encuentran en la industria. Este software en 3D incluye gráficos de alta calidad y sonido, ofreciendo un entorno bastante realista. Estas y otras cualidades hacen que este software de automatización industrial sea una herramienta práctica para la formación de estudiantes y profesionales.

El uso más común de FACTORY I/O es que sirve como plataforma de capacitación de PLCs, debido a que estos controladores son los que se utilizan de manera más frecuente en aplicaciones industriales.

Para la creación de un nuevo proyecto FACTORY I/O ofrece dos opciones para hacer el montaje de una escena de automatización las cuales son New y Scenes (Figura 14).

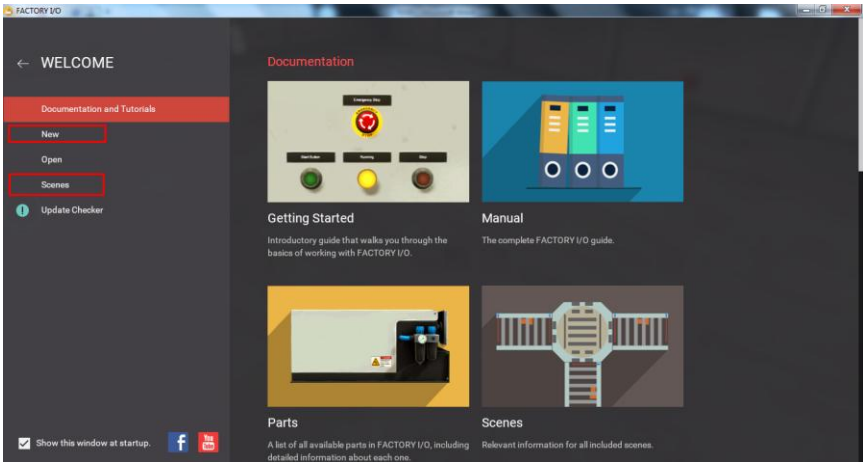

Figura 14 Selección del proyecto

Si se escoge *New*, aparecerá un espacio de trabajo vacio para poder ir creando la escena industrial que se desee. En esta opción se pueden encontrar una gran variedad de elementos como sensores, estaciones, operadores, entre otros (figura 15). Al elegir *Scenes*, aparecerán escenas comunes en aplicaciones industriales las cuales pueden ser editadas de acuerdo a la necesidad del programador como se muestra en la figura 16.

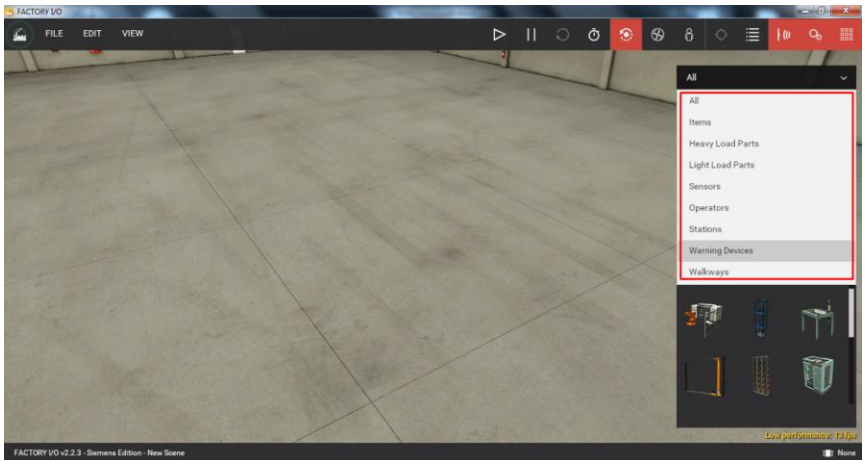

Figura 15 Proyecto en blanco

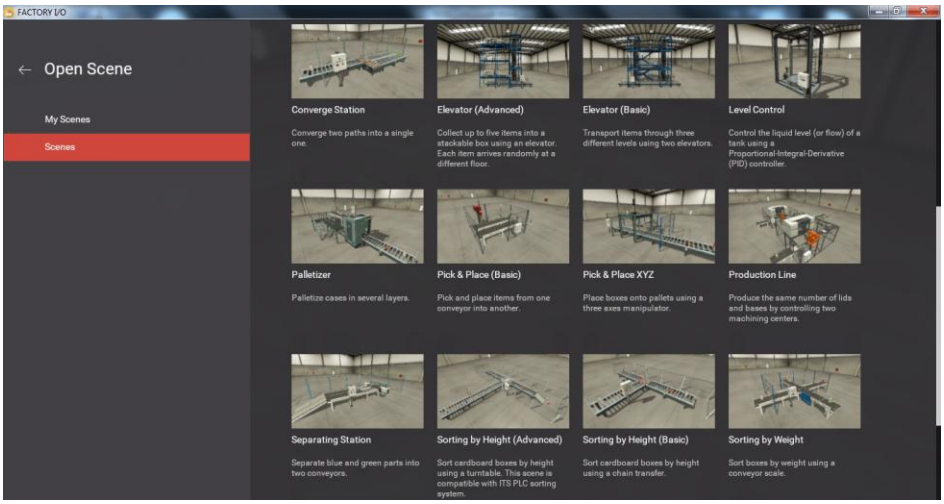

Figura 16 Escenas de trabajo

Los dos software descritos anteriormente no solo sirven para automatizar procesos combinacionales o secuenciales sino también para crear controladores por retroalimentación gracias a los lenguajes de alto nivel que posee CoDeSys.

3. Sistema de control retroalimentado

Se conoce como sistema de control retroalimentado a un sistema que mantiene una relación ya definida entre una señal de salida y una entrada puesta como referencia. Donde luego estas dos señales se comparan entre si y se utiliza la diferencia entre ellas como medio de control.

En la figura 17 se puede ver la estructura general de un control retro alimentado. Primeramente, se debe tener un valor deseado de la variable de proceso, este punto es conocido en sistemas de control como SetPoint. Luego se encuentra la planta, que representa todo el sistema que se le desea aplicar la estrategia de control. Afuera de este sistema se encuentra la señal de salida que es la variable que se desea controlar. Dicha variable se retroalimenta a través del sensor y se compara con el punto de control y si existe algún error se activa el actuador para llevar dicho error a cero y así controlar el sistema; todo este proceso de medir y activar el actuador lo hace el controlador.

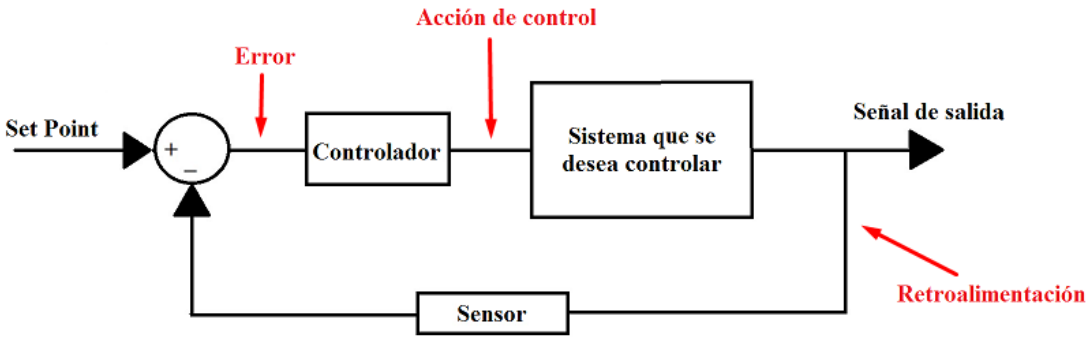

Figura 17 Diagrama de bloques de un control retroalimentado

Los controladores se pueden dividir en dos tipos, los cuales son: convencional y no convencional. Dentro del control convencional se encuentran dos teorías, la teoría de control clásico y la teoría de control moderna. En la teoría de control clásico se diseñan controladores basados en el análisis de frecuencia como diagrames de Bode. Y la teoría de control moderna es la que se basa en retroalimentación de estados.

3.1. Control convencional

3.1.1. Teoría de control clásico

Dentro de la teoría de control clásico se puede encontrar el controlador PID y existen tres tipos: Básico, I-PD y P-ID. En el controlador básico PID se encuentran las acciones de control proporcional, integral y derivativa sobre la señal de error directamente. En la I-PD se desea que la señal de acción no genere cambios bruscos, para esto la señal proporcional y derivativa se conectan a la señal de retroalimentación y nada más queda la acción integral sobre la señal del error. Por último, se tiene el controlador PI-D, donde esta versión sirve para evitar una "patada derivativa" cuando existe un cambio brusco en el error o en la señal de referencia, por lo tanto, para corregir la acción derivativa se conecta directamente en la señal de retroalimentación mientras que la acción proporcional e integral se encuentran sobre la señal de error.

También existen otras técnicas tales como el compensador en adelanto, donde éste sirve para acelerar la convergencia hacia la posición deseada. Otro compensador importante es el compensador en retardo ya que esta acción reduce o elimina el error en estado estable. Por último

se encuentra un hibrido entre los dos compensadores mencionados anteriormente el cual se conoce como compensador de retardo y adelanto; este compensador acelera la estabilización de la posición deseada y a su vez reduce el error de estado estable.

3.1.2. Teoría de control moderno

Por otra parte se encuentra la teoría de control moderno, donde también se pueden encontrar varias técnicas de control. Entre ellas se pueden destacar la retroalimentación con observadores y la retroalimentación de estado. La técnica de retroalimentación con observadores se usa en el caso de que no se conozca todo el estado del sistema, siendo así, se pueden poner observadores para estimar parte del estado y así formar la retroalimentación. En el control por retroalimentación de estado, se toma el estado del sistema que se desea controlar y cada uno de los estados se multiplica por una ganancia, luego, estas señales multiplicadas las retroalimenta hacia la entrada. Una gran ventaja de estos controladores es que se diseñan las ganancias de modo tal que los polos del sistema se puedan asignar donde se desee.

3.2. Control no convencional

3.2.1. Controlador difuso

Esta técnica de control ha ganado mucha popularidad debido a la facilidad en el diseño de controladores, ya que se pueden controlar sistemas no lineales demasiado complejos los cuales su modelado matemático es muy difícil y en ocasiones imposibles de describir. Este tipo de controladores tienen una ejecución más precisa con modelos complejos ya que usando métodos clásicos convencionales o lógica bivalente el controlador en su mayoría los puede ejecutar de manera errónea.

Debido a que no se requiere conocer el modelo matemático que describe la dinámica del sistema que se desea controlar, se pueden encontrar las siguientes ventajas:

- No se requiere identificar el sistema
- No se necesita aproximar el modelo
- No se necesita linealizar el modelo matemático del sistema
- No importa si se tienen múltiples entradas y salidas.

En la Figura 18 se muestran las etapas del controlador difuso, posteriormente se hace una breve descripción de cada una de ellas.

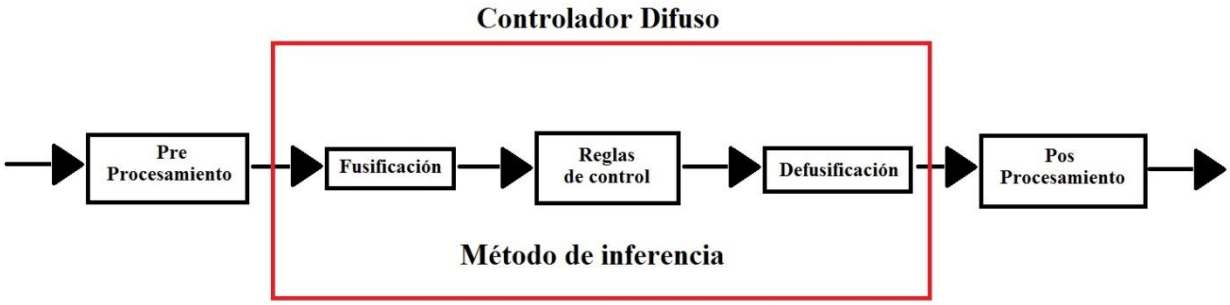

Figura 18 Etapas del controlador difuso

Pre-procesamiento: Sirve para acondicionar las señales que entran antes de introducirlas al procesador digital.

Fusificación: método por el cual se toman las magnitudes de las señales que entran y se convierten en una cantidad difusa.

Reglas de control: son reglas lingüísticas que definen como se debe controlar el sistema.

Método de inferencia: es un algoritmo en el que se guía a la computadora para inferir en la conclusión a partir de las señales que entran y las reglas de control.

Defusificación: Toma un conjunto difuso y lo convierte en una cantidad certera (un único numero) para así tomar la acción de control adecuada para modular o compensar los errores del sistema.

Pos-procesamiento: genera la señal de control a partir de la cantidad defusificada acondicionándola al actuador.

Para el modelado de estos controladores, es necesario comprender la matemática difusa. Este tipo de lógica se adapta mas al mundo real, ya que adopta la forma del ser humano al comunicarse

comprendiendo cuantificadores de cualidad como “mucho”, “muy”, “poco”, entre otros. En el siguiente capítulo se explicará algunos conceptos de la lógica difusa enfocada en el diseño de controladores.

4. Lógica difusa

La manera en la que piensa el ser humano es innegablemente, difusa. Los constantes cambios que percibimos en el mundo no se pueden definir por solo dos sentencias (falso o verdadero, grande o pequeño, mucho o poco) es por esto que el ingeniero electricista y matemático Lotfi Asker Sadeh revolucionó el mundo de la inteligencia artificial con su publicación en 1965 “Fuzzy Sets” proporcionando maneras simples de resolver problemas mediante el uso de reglas lógicas empleadas cada vez más en la industria y que son utilizadas para el control de procesos que matemáticamente son demasiado difíciles de modelar.

4.1. Conjuntos difusos

Los conjuntos difusos son funciones que no tienen sus fronteras estrictamente definidas, a diferencia de los conjuntos clásicos que tienen una sola frontera de manera vertical, donde si pertenece al conjunto es 1 y si no pertenece es 0. Es decir, en los conjuntos clásicos un elemento puede pertenecer a una sola clase, por lo tanto hay exclusión inmediata (Figura 19). Se puede decir entonces que los conjuntos clásicos se definen bajo la siguiente circunstancia:

Sea Y un conjunto que pertenece al dominio de X. Una función característica del conjunto Y, alcanza el valor $\mu_Y(x) = 1$ si $x \in Y$, y $\mu_Y(x) = 0$ si $x \notin Y$, $\mu: X \rightarrow \{0, 1\}$.

En los conjuntos difusos no existe la exclusión nombrada en los conjuntos clásicos, la mayoría de elementos pertenecen en cierta medida a dos o más conjuntos con diferente valor de membrecía, es decir un elemento puede pertenecer a más de una clase (Figura 20). Se puede decir entonces que los conjuntos difusos están definidos de la siguiente manera:

Sea Z un conjunto del dominio X. Una función de pertenencia $\mu_Z(x)$ del conjunto Z es una función que asigna valor o grado de membrecía a cada $x \in Z$, $\mu: X \rightarrow [0,1]$.

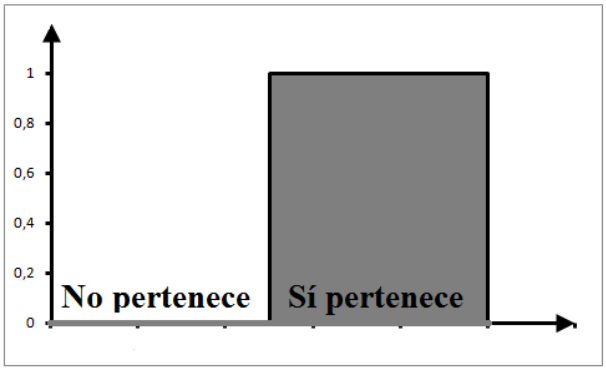

Figura 19 Valores de membrecía para un conjunto clásico

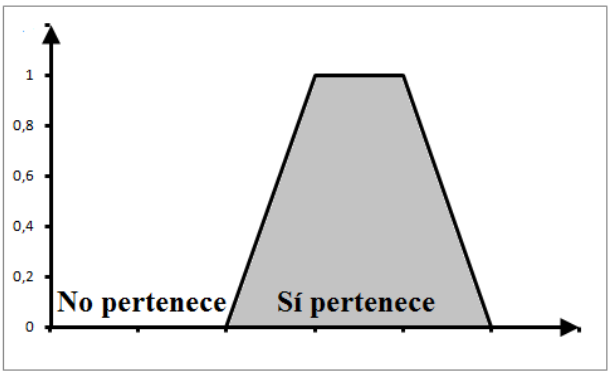

Figura 20 Valores de membrecía para un conjunto difuso

A continuación se muestra un ejemplo de un conjunto difuso junto con su notación matemática.

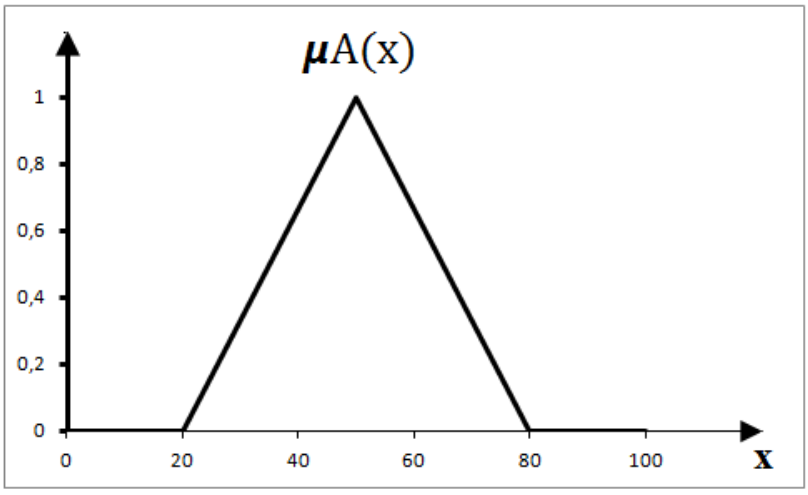

Figura 21 Conjunto difuso

La expresión matemática de este conjunto difuso viene dada de la siguiente manera:

$$\mu A(x) = \begin{cases} 0; & si\ x < 20 \\ \dfrac{x-20}{30}; & si\ 20 \leq x \leq 50 \\ \dfrac{80-x}{30}; & si\ 50 \leq x \leq 80 \\ 0; & si\ x > 80 \end{cases}$$

4.2. Funciones de membrecía

Las funciones de membrecía están formadas por un núcleo, una base y fronteras. En la figura 22 se muestra el conjunto difuso A con las partes que lo compone. En el núcleo se puede observar de manera gráfica que es la región donde los elementos dentro del conjunto tienen un valor de uno, es decir μA(x) = 1. La base es la región que comprende todo el conjunto, esto significa que ningún elemento del conjunto es igual a cero μA(x) ≠ 0 y por último se tienen las fronteras, que son valores que pertenecen al conjunto pero no tienen total pertenencia a él y por esta razón no son cero pero tampoco uno, es decir 0 < μA(x) < 1. Estas funciones de membrecía siempre van a tener un dominio llamado en lógica difusa universo de discurso y puede estar formado por todos los números reales, a diferencia de la imagen, que solo puede obtener valores reales entre cero y uno.

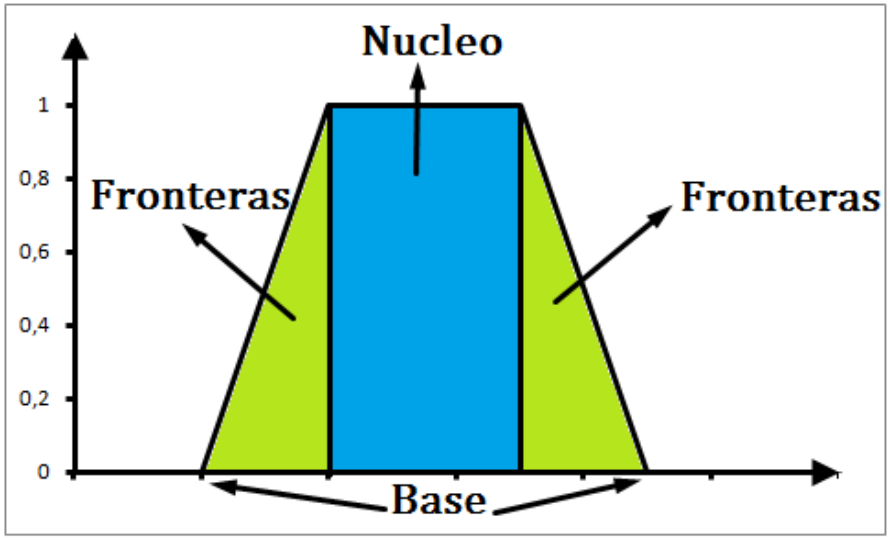

Figura 22 Partes de una función de membrecía

Las funciones de membrecía pueden tener diferentes formas. A continuación en las figuras 23, 24 y 25 se muestran las más utilizadas en el desarrollo de controladores difusos.

Función trapezoidal

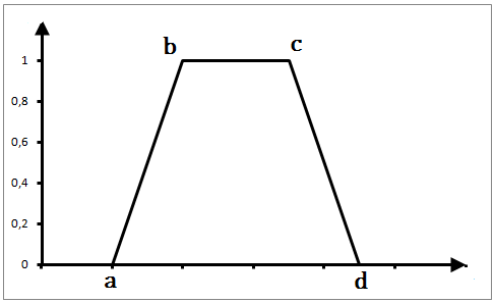

Figura 23 Función trapezoidal

$$\pi(x,a,b,c,d)\begin{cases} 0;\ si\ x < a \\ \dfrac{x-a}{b-a};\ si\ a \le x \le b \\ 1; si\ c \le x \le d \\ 0; si\ x > d \end{cases}$$

Función triangular

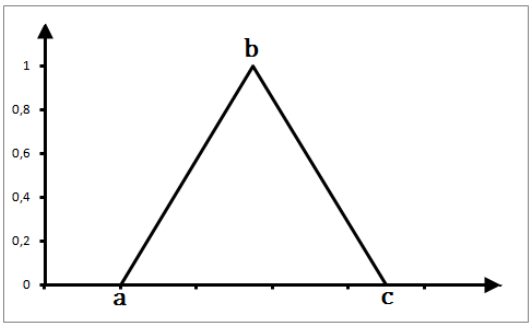

Figura 24 Función triangular

$$\wedge(x,a,b,c)\begin{cases}0; si\ x<a\\ \dfrac{x-a}{b-a}; si\ a\le x\le b\\ \dfrac{c-x}{c-b}; si\ b\le x\le c\\ 0; si\ x>c\end{cases}$$

Función Singleton

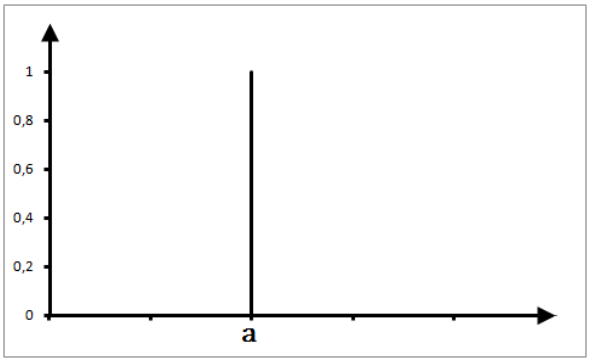

Figura 25 Función Singleton

$$\mu_A(x)=\begin{cases}1; si\ \mu=a\\ 0; si\ \mu\neq a\end{cases}$$

4.3. Operaciones entre conjuntos difusos

Como se observó en el capitulo anterior, los conjuntos clásicos podrían tratarse como un caso especial de los conjuntos difusos, por esta razón se partirá de la teoría clásica para mostrar las consideraciones necesarias para llegar a las operaciones entre conjuntos difusos.

4.3.1. Intersección

En la lógica clásica se parte del hecho de tener dos conjuntos A y B. La intersección que es un conjunto C, está formado por todos los elementos que están contenidos en A y B (Figura 26).

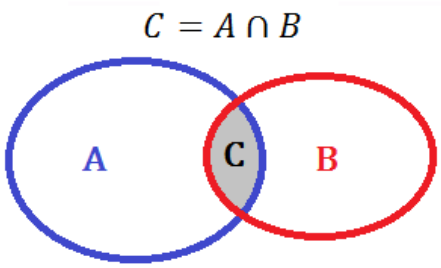

Figura 26 Intersección entre conjuntos clásicos

Partiendo del análisis de la lógica clásica. Gráficamente se puede observar que para realizar la intersección entre los conjuntos difusos A y B y generar un conjunto difuso C, los elementos que están contenidos en A y B son los valores mínimos para cada punto del universo de discurso (Figura 27).

$$\mu_C(x) = \min[\mu_A(x), \mu_B(x)] = \mu_A(x) \wedge \mu_B(x) \; para \; \forall x \in X$$

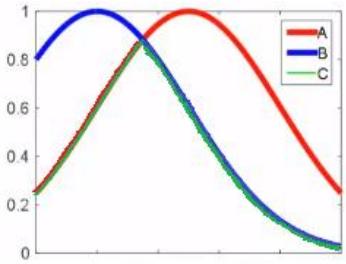

Figura 27 Intersección entre conjuntos difusos

4.3.2. Unión

Se tienen dos conjuntos A y B. La unión entre ellos es un conjunto C con todos los elementos contenidos en ambos conjuntos (Figura 28).

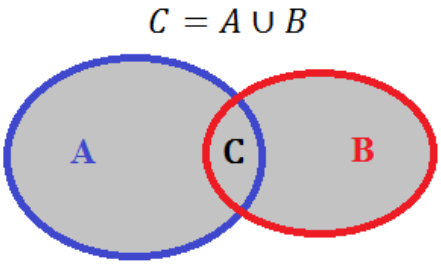

Figura 28 Unión entre conjuntos clásicos

Teniendo un punto de partida desde la lógica clásica. Gráficamente se puede observar que para hacer la unión entre los conjuntos difusos A y B y generar un conjunto difuso C, todos los valores que están contenidos en las funciones son los valores máximos para cada punto del universo de discurso como se muestra en la figura 29.

$$\mu_C(x) = \max[\mu_A(x), \mu_B(x)] = \mu_A(x) \vee \mu_B(x) \; para \; \forall x \in X$$

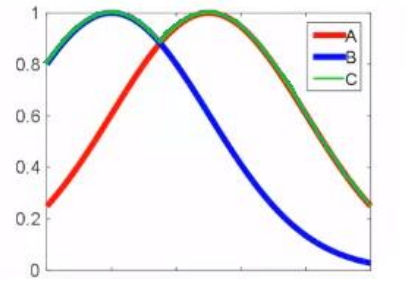

Figura 29 Unión entre conjuntos difusos

4.3.3. Complemento

Se tiene un conjunto A que está en un universo de discurso X, el complemento de A es todos los valores que están en el dominio de X pero que no se encuentran en A (Figura 30).

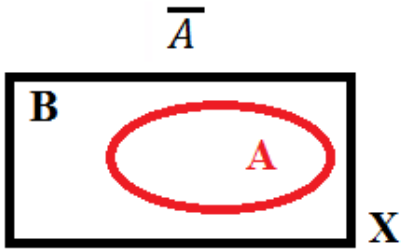

Figura 30 Complemento de un conjunto clásico

En los conjuntos difusos Sadeh encontró una expresión válida para conjuntos certeros y difusos la cual se muestra en la figura 31.

$$\mu_{\sim A} = 1 - \mu_A \; para \; \forall x \in X$$

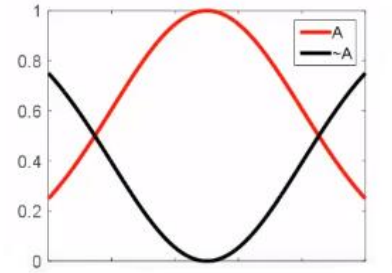

Figura 31 Expresión y gráfica del complemento de un conjunto difuso

4.3.4. Relación

En conjuntos certeros, una relación es una correspondencia entre dos conjuntos, el primer conjunto se conoce como dominio y al segundo conjunto se le llama imagen, donde a cada elemento del dominio se le atribuye al menos una imagen (figura 32). Si a cada elemento del dominio se les asigna una única imagen, a esa relación se le llama función (Figura 33).

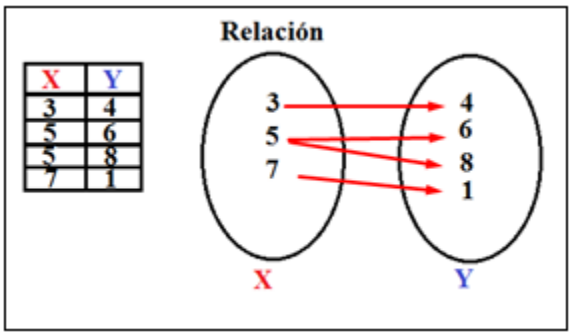

Figura 32 Relación entre conjuntos certeros

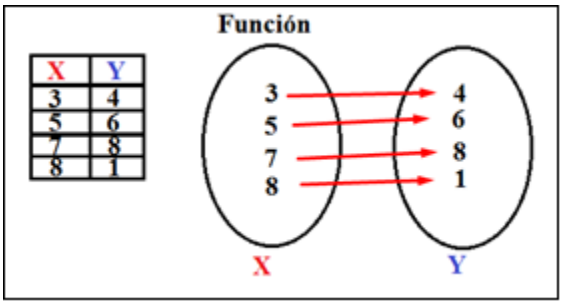

Figura 33 Función entre conjuntos certeros

Como Se observo en las figuras 32 y 33, esas son las formas básicas de representar las relaciones. También existe una forma poco convencional y es de manera matricial (figura 34).

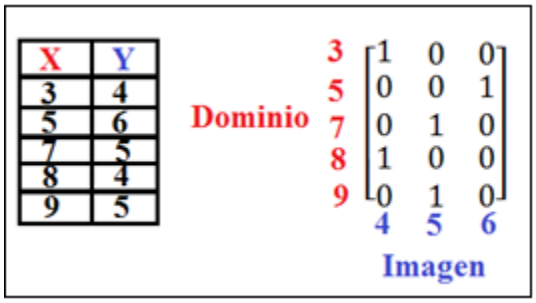

Figura 34 Forma matricial para representar relaciones

Se pone un cero (0) en la matriz cuando no hay relación y un uno (1) que indica que hay relación entre los pares de números.

Esta forma matricial también puede representar una relación difusa $\mathcal{R}$ como se muestra en la figura 35, con la diferencia que los elementos de la matriz pueden estar comprendidos con cualquier número real entre cero y uno. También se hace la aclaración que un elemento del dominio puede tener diferente grado de membrecía para varias imágenes, por esta razón nunca existirán funciones difusas.

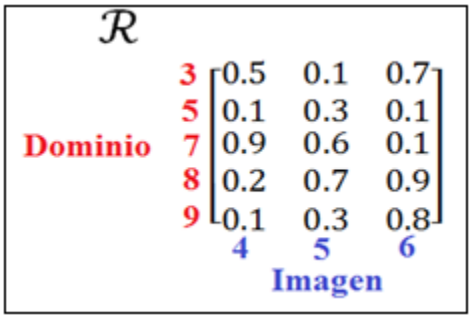

Figura 35 Relación difusa expresada de manera matricial

4.3.5. Producto cartesiano

De forma didáctica se muestra un ejemplo para mayor entendimiento del producto cartesiano en el caso certero.

Se tienen dos conjuntos X y Y, el conjunto X posee todos los reales entre cero y diez y el conjunto Y está formado por todos los reales entre cero y ocho (Figura 36).

Figura 36 Conjuntos

Se puede observar en la figura 37 que el producto cartesiano entre el conjunto X y Y es un plano con todos los posibles pares de números entre X y Y.

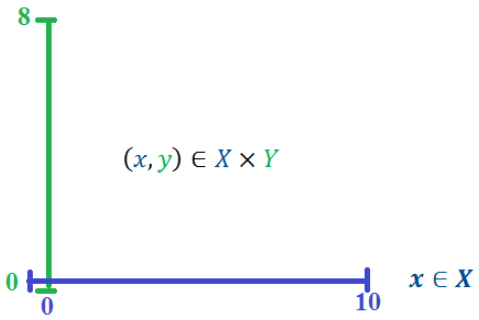

Figura 37 Producto cartesiano entre dos conjuntos

Lo nombrado anteriormente se puede extender al caso difuso. Como los conjuntos difusos van a tener valores de membrecía entre cero y uno se necesita entonces una definición para calcular los pares ordenados del producto cartesiano, para esto, Sadeh determinó que el producto cartesiano es una relación entre dos conjuntos y que cada par ordenado debe estar contenido en ambos, como ya se demostró, esta operación es la intersección, la cual está definida como el mínimo entre las funciones de membrecía.

$$\mu_{A\times B}(x,y) = \min\left(\mu_A(x), \mu_B(y)\right)$$

A continuación se muestra el siguiente ejemplo: Se desea calcular el producto cartesiano entre los conjuntos difusos A y B, el conjunto A esta representado por $A = \left\{\frac{0}{0} + \frac{0.1}{10} + \frac{0.5}{20} + \frac{0.7}{30} + \frac{1}{40} + \frac{1}{50}\right\}$ y $B = \left\{\frac{0.6}{1} + \frac{0.8}{2} + \frac{0.9}{3} + \frac{1}{4}\right\}$

$$\mathcal{R} = A \times B = \begin{array}{c} 0 \\ 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6 & 0.7 & 0.7 & 0.7 \\ 0.6 & 0.8 & 0.9 & 0.9 \\ 0.6 & 0.8 & 0.9 & 0.9 \end{bmatrix} \begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}$$
$$1 \quad 2 \quad 3 \quad 4$$

Figura 38 Matriz de relación difusa

4.3.6. Composición entre relaciones difusas

A partir del siguiente ejemplo, se explica la composición entre funciones. Se asumen dos funciones g y f, las cuales tienen tres conjuntos X, Y y Z como se muestra en la figura 39.

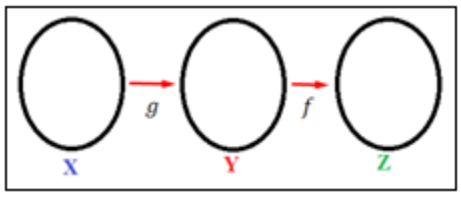

Figura 39 Función compuesta

Se observa una función g en la cual Y depende de X y una función f donde el conjunto Z depende de Y. La composición busca conocer la relación directa entre X y Z, esto se expresa matemáticamente de la siguiente manera.

$$Z = f(y) = f\big(g(x)\big) \therefore f \circ g(x)$$

En el caso difuso, las relaciones contienen valores de membrecía para cada uno de los pares de números que relacionan. Para hallar el valor de membrecía que corresponde a la relación directa se usa la siguiente expresión.

$$\mu_{\mathcal{R}1 \circ R2}(x, z) = \vee_y \left[\mu_{\mathcal{R}1}(x, y) \wedge \mu_{\mathcal{R}2}(y, z)\right]$$

Para dar claridad se resuelve el siguiente ejemplo donde se consideran las relaciones $\mathcal{R}1$ y $\mathcal{R}2$ y se desea hallar la composición entre ellas. $\mathcal{R}1$ es el resultado del producto cartesiano entre A y B y $\mathcal{R}2$ es el producto cartesiano del conjunto B con C.

$$\mathcal{R}1 = A \times B = \begin{bmatrix} 0.3 & 1 & 0.1 \\ 0.1 & 0 & 0.6 \\ 0.9 & 0.5 & 0.7 \end{bmatrix} \qquad \mathcal{R}2 = B \times C = \begin{bmatrix} 1 & 0.7 & 0.5 & 0.4 & 0 \\ 0.5 & 0.6 & 0.7 & 0.3 & 0.3 \\ 0.6 & 1 & 0.5 & 0.2 & 0.8 \end{bmatrix}$$

Para hallar el término ij de la matriz $\mathcal{R}1 \circ \mathcal{R}2$ se toma la fila i de la relación difusa $\mathcal{R}1$ y la columna J de la relación $\mathcal{R}2$. Se hace una comparación elemento a elemento de la fila con la columna correspondiente y se hallan los valores mínimos para luego calcular el máximo entre ellos. Por ejemplo, para calcular el elemento de la fila 1, columna 1 de la matriz $\mathcal{R}1 \circ \mathcal{R}2$ se hace el siguiente procedimiento (Figura 40).

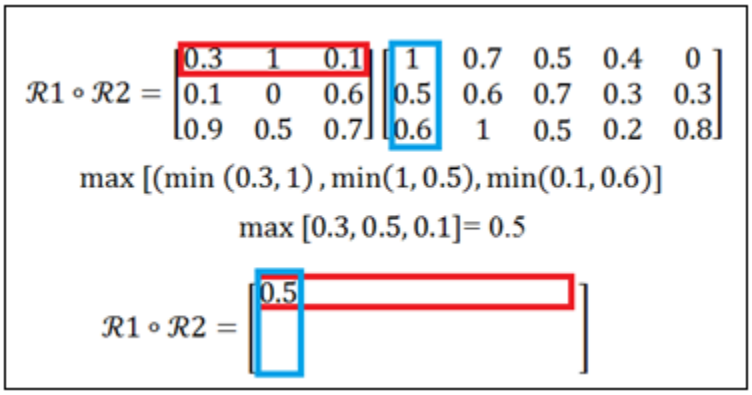

Figura 40 Procedimiento para hallar los elementos de la composición entre relaciones

Después de hacer el mismo procedimiento para cada uno de los elementos, se obtiene la siguiente matriz.

$$\mathcal{R}1 \circ \mathcal{R}2 = \begin{bmatrix} 0.5 & 0.6 & 0.7 & 0.3 & 0.8 \\ 0.6 & 0.6 & 0.5 & 0.2 & 0.6 \\ 0.9 & 0.7 & 0.5 & 0.4 & 0.7 \end{bmatrix}$$

4.3.7. Proposición clásica vs proposición difusa

En la proposición clásica se pueden encontrar dos sentencias, las cuales pueden ser verdaderas (1) o falsas (0). A continuación en la tabla 1 se muestran algunas operaciones lógicas que se pueden hacer entre proposiciones.

p	q	$p \wedge q$	$p \vee q$	$\check{p}$	$p \rightarrow q$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	0	1

Tabla 1 Operaciones lógicas

Conjunción: para que sea verdadera, es necesario que ambas proposiciones sean verdaderas y está determinada por "y ".

Disyunción: esta operación lógica es lo contrario de la conjunción, donde solo va a ser falsa cuando ambas sean falsas. Se simboliza con un "V" y está determinada por "O".

Negación: es un cambio de estado donde se tiene la sentencia negada, es decir si la sentencia es falsa se vuelve verdadera, y si es verdadera se vuelve falsa.

Implicación: esta operación tiene dos partes, un antecedente y un consecuente o conclusión. Solo en el caso en que la consecuencia sea falsa la implicación también será falsa.

Las tres primeras operaciones son elementales, a partir de ahí se pueden sacar más operaciones. Por ejemplo, la implicación que se nombró anteriormente tiene la siguiente equivalencia.

$$\boldsymbol{p \rightarrow q = \tilde{p} \vee q}$$

Por otra parte, la proposición difusa es una sentencia que puede tomar valores intermedios entre cero (0) y uno (1). Se toma entonces la tabla de verdad perteneciente a la lógica clásica y se empiezan a definir las operaciones lógicas.

Conjunción: para calcular la veracidad de dos proposiciones difusas se debe sacar el mínimo de estos dos valores de veracidad.

$$\boldsymbol{V(p \wedge q) = min(V(p), V(q))}$$

Disyunción: se verifica en la tabla de verdad que se debe hacer el máximo entre los valores de veracidad.

$$\boldsymbol{V(p \vee q) = max(V(p), V(q))}$$

Negación: se hace a través de un complemento a uno y se da por la siguiente expresión.

$$\boldsymbol{V(\tilde{p}) = 1 - V(p)}$$

Como en la lógica clásica, estas también son las tres operaciones elementales. Para la implicación se puede descomponer en términos de las operaciones básicas y se define de la siguiente manera.

$$\boldsymbol{V(p \rightarrow q) = V(\tilde{p} \vee q) = \max\left(1 - V(p), V(q)\right)}$$

A partir de las operaciones elementales se pueden hacer proposiciones compuestas para la lógica clásica y difusa. Una proposición muy importante para el diseño de controladores es el MODUS PONENDO PONENS

$$\boldsymbol{p \wedge (p \rightarrow q) \rightarrow q}$$

El Modus Ponendo Ponens que en latín significa el modo que afirmando afirma, es una regla de inferencia que saca una conclusión de proposiciones o premisas y así guiar una argumentación valida. Se observa en la ecuación # como está en términos condicionales uno depende del otro, es decir “si P entonces Q y tienes P, tendrás Q” para mayor entendimiento se muestra el siguiente ejemplo.

Si la física estudia la conducta de los cuerpos, entonces la física estudia la velocidad.

La física estudia la conducta de los cuerpos

Por lo tanto, la física estudia la velocidad.

4.3.8. Modus Ponens Difuso

Se constituye por dos premisas las cuales son un hecho y una regla. Con estas dos premisas se obtiene una conclusión (Figura 41).

Premisa 1 (Hecho)	x es A'
Premisa 2 (Regla)	si x es A, entonces y es B
Conclusión	y es B'

Figura 41 Modus Ponendo Ponens Difuso

A', A y B son términos lingüísticos representados por conjuntos difusos los cuales están formados por funciones de membrecía. x y y son variables lingüísticas y pueden representar cualquiera de las funciones. Este modus ponen difuso sirve para poder concluir razonamientos en la computadora; para este proceso se necesita una interpretación matemática la cual está dada por la siguiente expresión.

$$\frac{\begin{matrix} A' \\ \mathcal{R} = A \times B \end{matrix}}{B' = A' \circ \mathcal{R}}$$

5. Diseño de un controlador difuso

En este capítulo se hace una descripción de los pasos a seguir para el diseño de un controlador difuso, introduciendo definiciones como variables lingüísticas, reglas de control, método de inferencia y defusificación. En el método de inferencia se muestran las operaciones necesarias tomadas del capítulo 4 de lógica difusa y se demostrará el método de inferencia de Mamdani utilizado en este trabajo de grado para la implementación de un controlador difuso, por último se muestran algunos de los métodos más utilizados para defusificar.

5.1. Variables lingüísticas

Para diseñar un controlador difuso se deben definir las variables que entran y salen del controlador y se le conocen como variables lingüísticas, las cuales tienen un universo de discurso que es el rango donde se pueden sensar las señales de control. Estas variables están definidas en términos lingüísticos y se representan en formas de funciones de membrecía, donde cada una de estas relaciones representa un cuantificador de cualidad (muy pequeño, pequeño, cero, muy

grande, muy caliente, poco frio) y es así como se definen los errores o las acciones de los actuadores.

5.2. Reglas de control

Son reglas que están compuestas bajo una estructura SI y ENTONCES (*IF and THEN*). Estas reglas tratan de acercarse a las formas en que se plantea el ser humano para tomar decisiones bajo unas condiciones asignadas. Se puede decir entonces que estas reglas son la inteligencia del controlador difuso y depende de ellas el éxito al momento de controlar el sistema.

***IF** A and B **THEN** C*

SI A **y** B **ENTONCES** C

Para determinar cuántas reglas difusas necesita el controlador, es necesario identificar la cantidad de funciones de membrecía que componen cada una de las variables de entrada (en caso de que sean dos o más) y se obtendrá las posibles combinaciones entre las funciones de membrecía de cada conjunto difuso. La cantidad de funciones de membrecía pertenecientes al conjunto difuso de salida no depende de la cantidad de variables de entrada ni de sus respectivas funciones. Para comprender más a fondo se muestra el siguiente ejemplo.

Se tienen dos variables de entrada y estas se definen como los conjuntos difusos A y B. Los conjuntos difusos A y B tienen 5 y 3 funciones de membrecía respectivamente, la cantidad de reglas que debe tener el controlador es las posibles combinaciones entre los elementos del conjunto difuso A con la cantidad de elementos del conjunto difuso B, por lo tanto, la cantidad de reglas **SI** y **ENTONCES** que tendría el controlador son 5x3= 15.

5.3. Método de inferencia de Mamdani

Se tiene un conjunto de reglas SI y ENTONCES formado por una entrada y una salida, donde X es la variable de entrada del controlador y Y es la salida. Existe entonces una variable lingüística

de entrada A compuesta por i funciones de pertenencia $(A1, A2 \ldots . Ai)$ y una variable lingüística de salida B compuesta con i funciones de membrecía $(B1, B2 \ldots . Bi)$. Las reglas mencionadas anteriormente se expresan de la siguiente manera.

$$\begin{array}{c} X\ es\ A' \\ SI\ X\ es\ A1, ENTONCES\ Y\ es\ B1\ (Regla\ 1) \\ SI\ X\ es\ A2, ENTONCES\ Y\ es\ B2\ (Regla\ 2) \\ \hline Y\ es\ B' \end{array}$$

Ecuación 5-1

Para calcular la salida B' se debe hacer una interpretación matemática del Modus Ponen Difuso mencionado en el capitulo anterior.

5.3.1. Interpretación matemática del método de inferencia de Mamdani

Para calcular B' se deben hallar los $B'i$ pertenecientes a cada regla difusa. Se parte entonces de la regla difusa uno y se generaliza para las demás reglas. La regla uno es la composición entre el hecho (A') y la relación difusa formada por el antecedente $A1$ y el consecuente $B1$. Ya obtenidos los conjuntos difusos $B'i$ se hace la unión entre ellos y se obtiene el conjunto difuso B'.

$$\begin{array}{c} A' \\ B'1 = A' \circ \mathcal{R}1 = A' \circ (A1 \times B1) \\ B'2 = A' \circ \mathcal{R}2 = A' \circ (A2 \times B2) \\ \hline B' = B'1 \cup B'2 \ldots \ldots \ldots \ldots \ldots \ldots B'i \end{array}$$

Ecuación 5-2

Todo este proceso matemático se puede simplificar y evitar la resolución de composiciones y productos cartesianos. Para esto, se toma como ejemplo calcular la función de membrecía de $B'i$ $(\mu_{B'i}(y))$. Este conjunto está dado por la composición entre el conjunto A' y una relación difusa $\mathcal{R}$. Como se demostró en el capitulo anterior, la composición está representada por el mínimo entre la funciones de membrecía de A' y $\mathcal{R}i$, donde $\mathcal{R}i$ es una relación que está entre los dominios

de X y Y. Por último, se hace un máximo tomando todos los valores respecto a X ya que es la variable que relaciona a estos conjuntos.

$$\mu_{B'i}(y) = \vee_x [\mu_{A'}(x) \wedge \mu_{\mathcal{R}i}(x, y)]$$

La relación $\mathcal{R}i$ se obtiene a partir del producto cartesiano, donde en el capitulo anterior se demostró que se puede resolver a través del mínimo entre la función de membrecía de Ai y Bi

$$\mu_{B'i}(y) = \vee_x [\mu_{A'}(x) \wedge \mu_{\mathcal{R}i}(x, y)] = \vee_x [\mu_{A'}(x) \wedge \mu_{Ai}(x) \wedge \mu_{Bi}(y)]$$

Se observa que la maximización es respecto a X, debido esto, el último término se puede sacar de la agrupación.

$$\vee_x [\mu_{A'}(x) \wedge \mu_{Ai}(x) \wedge \mu_{Bi}(y)] = \{\vee_x [\mu_{A'}(x) \wedge \mu_{Ai}(x)]\} \wedge \mu_{Bi}(y)\}$$

Ecuación 5-3

Se resuelve lo que hay dentro de las llaves partiendo del hecho que en los controladores difusos A' es una señal que evoluciona en el tiempo y en un tiempo especifico esta señal tiene un único valor, es decir, la entrada es un conjunto Singleton.

Considerando entonces que A' es un conjunto Singleton se recurre a un análisis grafico.

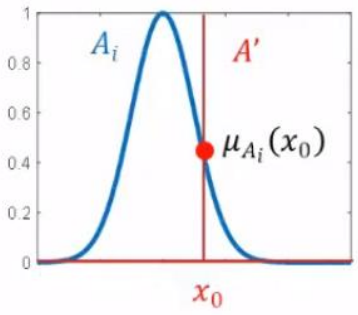

Figura 42 Valor mínimo entre la entrada del controlador y una función de membrecía

Se observa que A' es cero en todo el universo de discurso excepto en X_0 que es el dato que le entra al controlador y en ese punto la función de membrecía toma un valor de 1. Solo cuando X es igual a X_0, el mínimo va a ser entre 1 y el valor que tenga la función de membrecía Ai en X_0. Es decir $\mu_{A'}(x) \wedge \mu_{Ai}(x) = \mu_{Ai}(x_0)$. Se observa entonces en el término de la ecuación 5-3

$\vee_x [\mu_{A'}(x) \wedge \mu_{Ai}(x)]$ se debe hallar el máximo de los mínimos, pero como el resto de mínimos son cero, el máximo de los mínimos sigue siendo $\mu_{Ai}(x_0)$. Por lo tanto:

$$\vee_x [\mu_{A'}(x) \wedge \mu_{Ai}(x)] = \mu_{Ai}(x_0)$$

A esta evaluación se le conoce como valor de fusificación del antecedente Ai en X_0. La expresión 5-3 se reescribe como:

$$\mu_{B'i}(y) = \mu_{Ai}(x_0) \wedge \mu_{Bi}(y)$$

Ecuación 5-4

Al hacer esto para la regla i se puede hacer para cualquier regla, donde al final se debe hacer una maximización entre todos los conjuntos resultantes para hallar $\mu_{B'}(y)$.

$$\begin{array}{c} \mu_{A'}(x) \\ \mu_{B'1}(y) = \mu_{A1}(x_0) \wedge \mu_{B1}(y) \\ \mu_{B'2}(y) = \mu_{A2}(x_0) \wedge \mu_{B2}(y) \\ \vdots \qquad\qquad\qquad \vdots \\ \hline \mu_{B'}(y) = \mu_{B'1}(y) \vee \mu_{B'2}(y) \vee \ldots \vee \mu_{B'i}(y) \end{array}$$

Ecuación 5-5

A continuación se muestra el método de inferencia de Mamdani para dos entradas, teniendo entonces dos antecedentes y un consecuente y así generalizarlo para varias entradas.

$$\begin{array}{c} x\ es\ A'\ y\ y\ es\ B' \\ SI\ x\ es\ A1\ y\ y\ es\ B1, ENTONCES\ z\ es\ C1\ (Regla\ 1) \\ SI\ x\ es\ A2\ y\ y\ es\ B2, ENTONCES\ z\ es\ C2\ (Regla\ 2) \\ \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \hline z\ es\ C' \end{array}$$

Ecuación 5-6

El Modus Ponen Difuso se debe hacer entre el hecho y cada una de las reglas para generar un conjunto conclusión C'. Para esto, se debe tener en cuenta que el hecho define a x y y como A' y B' respectivamente y matemáticamente se representan como una relación difusa ($\mathcal{R}' = A' \times B'$) de la misma forma sucede con las reglas de control ($\mathcal{R}i = Ai \times Bi \times Ci$).

Se debe hacer entonces una composición difusa entre el hecho y la relación difusa proveniente de cada una de las reglas difusas.

$$C'i \ = \ \mathcal{R}' \circ \mathcal{R}i$$

Por último, para hallar el conjunto difuso de salida C' se unen todos los conjuntos hallados por cada regla difusa y se hace una unión.

$$C' = \ C'1 \cup C'2 \cup \dots \dots \cup C'i$$

La expresión matemática que define el método de inferencia para dos entradas es la siguiente:

$$\begin{array}{c} \text{X es A}' \text{ y Y es B}' \\ C'1 = (A' \times B') \circ (A1 \times B1 \times C1) \\ C'2 = (A' \times B') \circ (A2 \times B2 \times C2) \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ \hline C' = \ C'1 \cup C'2 \cup \dots \dots \cup C'i \end{array}$$

Ecuación 5-7

La inferencia de Mamdani hace el Modus Ponens a través de los productos cartesianos y composiciones. Al final se hacen las uniones entre los conjuntos conclusión de cada regla. Todo este proceso se puede simplificar gracias a que el conjunto A' y B' son conjuntos Singleton.

Se procede a calcular la función de membrecía de cualquiera de estas reglas.

$$C'i = (A' \times B') \circ (Ai \times Bi \times Ci) = \ \vee_{x,y} \left[\mu_{R'}(x,y) \wedge \mu_{\mathcal{R}i}(x,y,z)\right]$$

Se descomponen las relaciones difusas teniendo en cuenta que $\mathcal{R}'$ es el producto cartesiano entre A'y B' y un producto cartesiano se hace con el mínimo de sus valores de membrecía.

$$A' \times B' = \mu_{R'}(x,y) = \mu_{A'}(x) \wedge \mu_{B'}(y)$$

Esto sucede también con $\mathcal{R}i$ que proviene de un producto cartesiano triple.

$$Ai \times Bi \times Ci = \mu_{Ri}(x,y,z) = \mu_{Ai}(x) \wedge \mu_{Bi}(y) \wedge \mu_{Ci}(z)$$

Se hacen los respectivos reemplazo y queda la función de membrecía de Ci de la siguiente manera:

$$\mu_{C'i}(z) = \vee_{x,y}\ [\mu_{R'}(x,y) \wedge \mu_{\mathcal{R}i}(x,y,z)] = \vee_{x,y}\ [\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_{Ai}(x) \wedge \mu_{Bi}(y) \wedge \mu_{Ci}(z)]$$

Ecuación 5-8

Teniendo la función de membrecía $\mu_{C'i}(z)$ se hacen las siguientes consideraciones:

- Se debe hacer maximización respecto a X y Y pero no respecto a Z
- El valor de membrecía de Ai y A' solo depende de X
- El valor de membrecía de Bi y B' solo depende de Y

Bajo estas observaciones se puede agrupar de la siguiente manera

$$\mu_{C'i}(z) = \{\vee_x\ [\mu_{A'}(x) \wedge \mu_{Ai}(x)\} \wedge \{\vee_y\ [\mu_{B'}(y) \wedge \mu_{Bi}(y)\} \wedge \mu_{Ci}(z)$$

Teniendo en cuenta la ecuación 5-4 donde se demostró de manera grafica el valor de fusificación se puede llegar a la siguiente expresión:

$$\mu_{C'i}(z) = \mu_{Ai}(x_0) \wedge \mu_{Bi}(y_0) \wedge \mu_{Ci}(z)$$

Ecuación 5-9

El método de inferencia de Mamdani queda de la siguiente manera:

$$\begin{array}{c} \mu_{A'}(x)\ y\ \mu_{B'}(y) \\ \mu_{C'1}(z) = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0) \wedge \mu_{C1}(z) \\ \mu_{C'2}(z) = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0) \wedge \mu_{C2}(z) \\ \vdots \qquad\qquad \vdots \\ \hline \mu_{C'}(z) = \mu_{C'1}(z) \vee \mu_{C'2}(z) \vee \ldots \vee \mu_{C'i}(z) \end{array}$$

Ecuación 5-10

5.3.2. Interpretación gráfica del método de inferencia de Mamdani

$$\begin{array}{c} \mu_{A'}(x)\ y\ \mu_{B'}(y) \\ \mu_{C'1}(z) = \mu_{A1}(x_0) \wedge \mu_{B1}(y_0) \wedge \mu_{C1}(z) \\ \mu_{C'2}(z) = \mu_{A2}(x_0) \wedge \mu_{B2}(y_0) \wedge \mu_{C2}(z) \\ \vdots \qquad\qquad \vdots \\ \hline \mu_{C'}(z) = \mu_{C'1}(z) \vee \mu_{C'2}(z) \vee \ldots \vee \mu_{C'i}(z) \end{array}$$

Se considera una regla 1 conformada por tres funciones de membrecía donde $A1$ y $B1$ son los antecedentes de las regla y $C1$ es el consecuente. También se tiene una segunda regla donde A2 y B2 son los antecedentes y el consecuente es C2. Lo que se interprete para estas dos reglas es lo mismo que se haría para las i reglas que tenga el controlador (Figura 43).

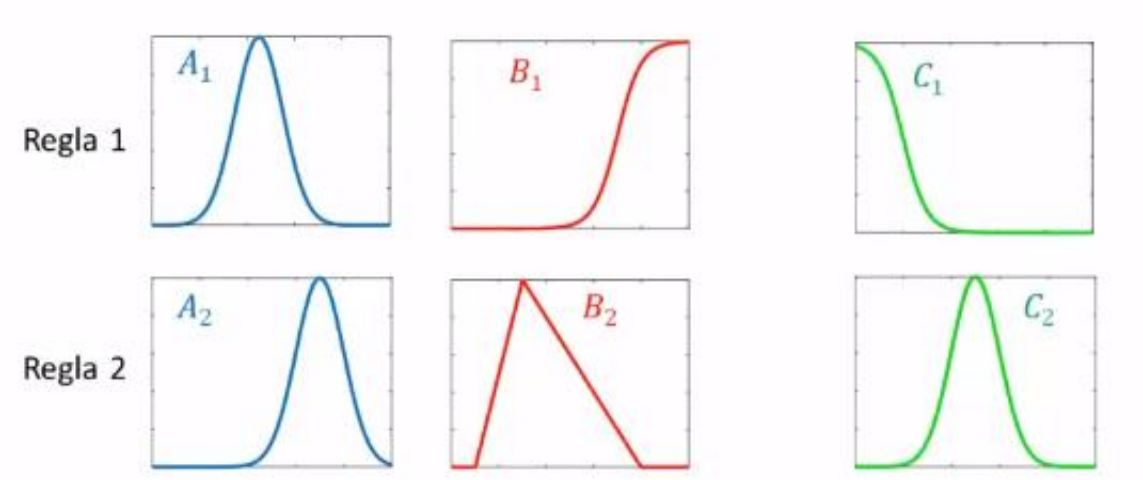

Figura 43 Reglas difusas del controlador

Al controlador llegan dos datos de entrada y son valores escalares los cuales son X_0 y Y_0 y lo que se debe hacer de acuerdo a las expresiones matemáticas es fusificar estos valores en las funciones de membrecía $A1$, $B1$, $A2$ y $B2$ (figura 44).

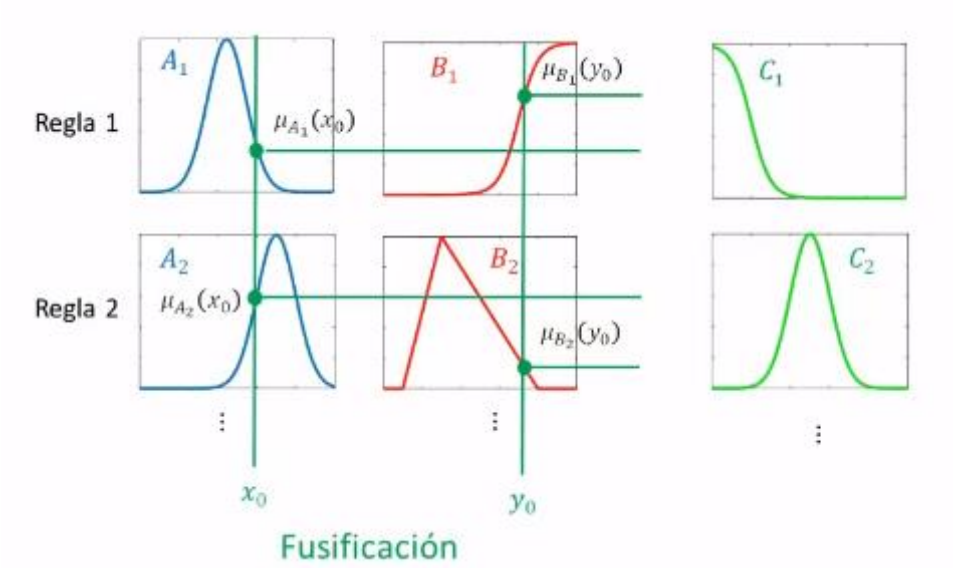

Figura 44 Valores de fusificación para cada función de membrecía de entrada

Como lo indica la expresión matemática del método de inferencia de Mamdani, se necesita calcular el mínimo entre $\mu_{A1}(xo)$ y $\mu_{B1}(yo)$, ya encontrado ese valor se debe sacar el mínimo entre el dato hallado y la función de membrecía $C1$ que hace parte del conjunto difuso de la salida. Todo lo que hace este procedimiento es cortar la función de salida a la altura del valor fusificado más pequeño de la regla, esta nueva función se conoce como $C'1$ (Figura 45). Estos pasos se repiten para las i reglas que posea el controlador.

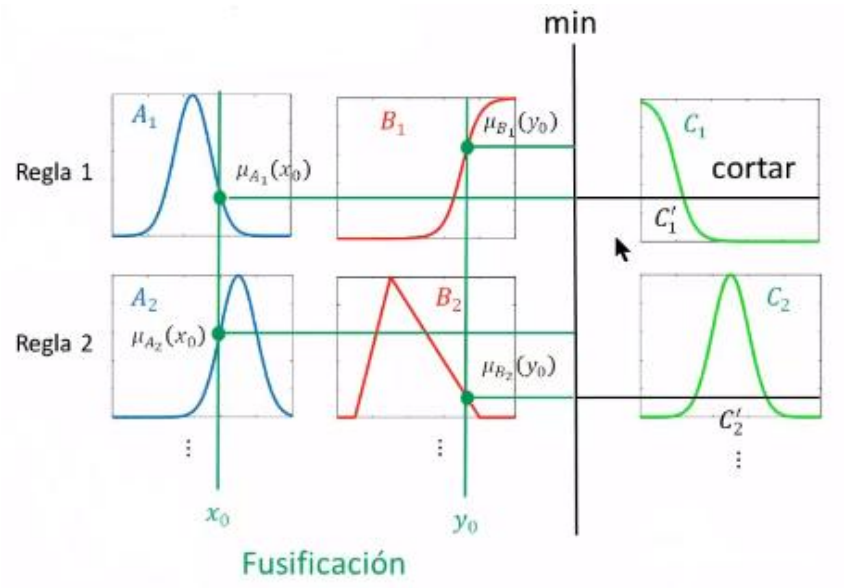

Figura 45 Funciones de salida cortadas a la altura del valor fusificado mas pequeño

Por último se debe calcular un conjunto C' (Figura 46) que es la unión de todos los conjuntos resultantes $C'i$.

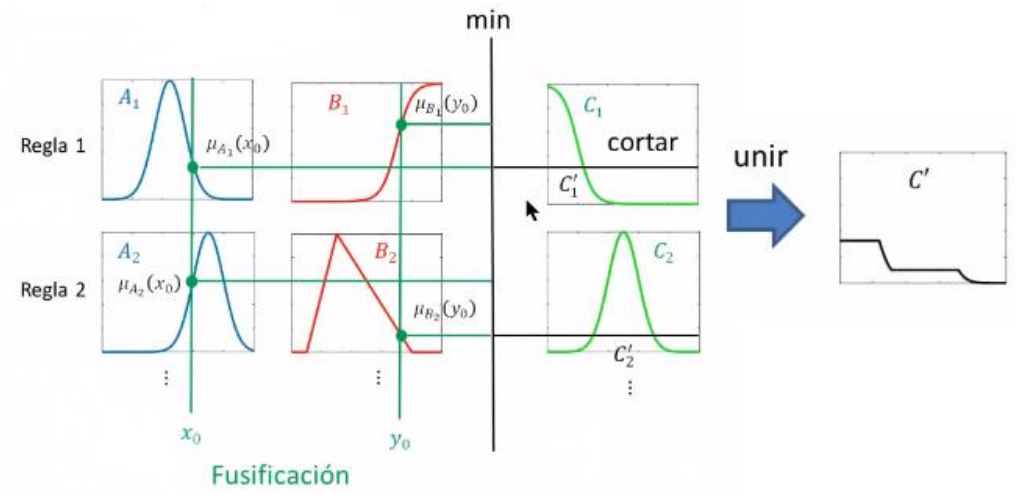

Figura 46 Unión de los conjuntos resultantes

Queda un conjunto C´ pero se debe tener en cuenta que a la salida del controlador solo puede salir un único valor escalar, para esto se debe hacer un proceso llamado defusificación.

5.4. Defusificación

Como se observa en la interpretación grafica del método de inferencia de Mamdani por lo general queda un conjunto difuso después de unir los conjuntos resultantes de cada una de las reglas. Al controlar el sistema no se pueden aplicar diferentes cantidades al actuador para realizar la acción de control y compensar el error, es por esto que se utiliza algún método de defusificación para así obtener un único valor escalar como se muestra en la figura 47.

Figura 47 Defusificación

A continuación se muestran varias formas de hallar un valor defusificado.

5.4.1. Método del centroide

Busca el centro geométrico o baricentro del conjunto resultante C'. Este método es el más usado ya que es el más rápido y exacto al momento de controlar, entre sus desventajas es la resolución de integrales, debido a esto se debe usar de manera extra un micro controlador o software donde se resuelva de manera práctica estos cálculos matemáticos (Figura 48).

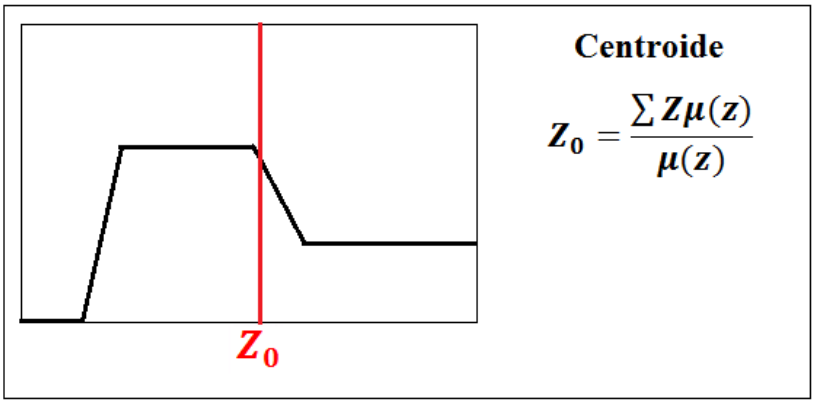

Figura 48 Método del centroide

5.4.2. Método de la Bisectriz

Busca dividir el área bajo la curva de la función de membrecía en dos regiones iguales a cada lado (Figura 49).

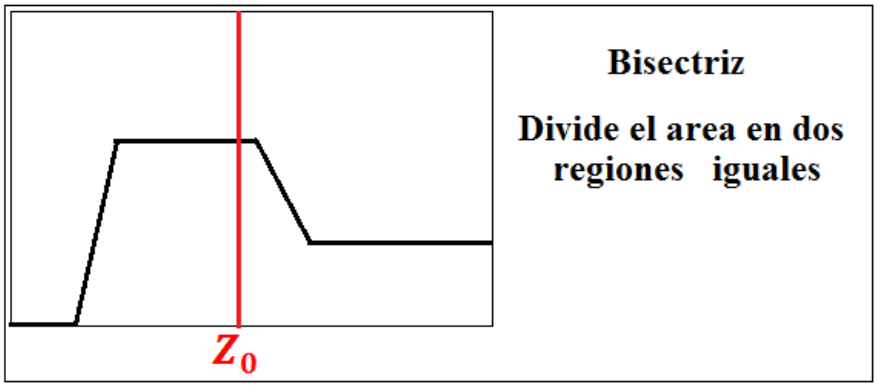

Figura 49 Método de la bisectriz

5.4.3. Método del Máximo central (MOM)

Este método encuentra el intervalo de valores de C´ para los cuales se maximiza la función de membrecía, una vez encontrado esos intervalos se debe calcular el valor de salida por medio de un promedio ponderado. Para que este método sea eficiente, se deben realizar las funciones de membrecía de forma simétrica a las entradas de cada conjunto difuso. Este método es una buena opción cuando se tienen lenguajes de programación con recursos limitados (Figura 50).

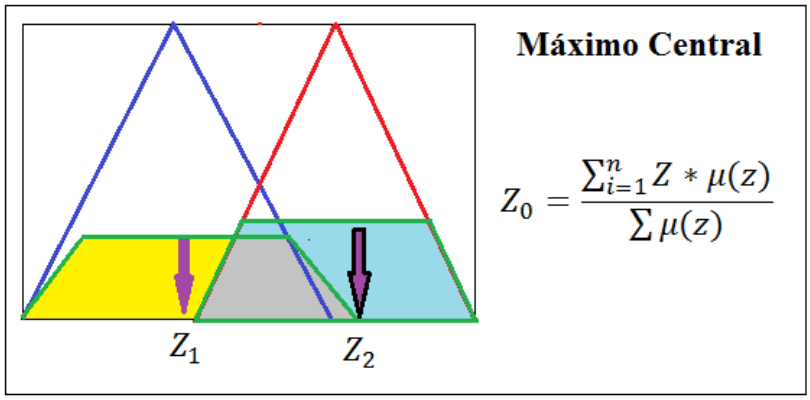

Figura 50 Método del máximo central

6. Objetivos

6.1. Objetivo general

- Implementar un controlador difuso en un controlador lógico programable utilizando al menos un lenguaje de programación normalizado según el estándar IEC 61131.

6.2. Objetivos específicos

- Estudiar los lenguajes de programación que pertenecen al estándar internacional IEC 61131-3.

- Estudiar los conceptos de lógica difusa, automatización y control necesarios para el desarrollo del proyecto.
- Desarrollar un controlador difuso.
- Programar el controlador difuso en un PLC utilizando al menos un lenguaje de programación normalizado en IEC 61131.
- Verificar la operación del controlador diseñado en un sistema real.

7. Planteamiento del problema

Para este proyecto de grado y en función del objetivo general se diseñó un ejercicio muy común en procesos automatizados que es el control de nivel de líquido en un tanque, este tipo de control tiene varias aplicaciones en la industria como sistemas de mezclado de sustancias, aplicación en circuitos de fluidos que entran y salen de un deposito, entre otros. Este ejercicio fue resuelto con el ánimo de obtener resultados concluyentes y alcanzar los objetivos de manera satisfactoria.

Con base en lo anterior se exponen los siguientes ejercicios: Se desea diseñar dos controladores difusos, el primero es un controlador proporcional difuso que solo posee una salida la cual es una válvula de llenado, el error de estado estable debe ser menor al 1% y en el momento de aplicar la estrategia de control no se puede tener sobre-impulsos ya que no existe un segundo actuador que pueda generar una oscilación que permita establecer el error. El segundo controlador es un proporcional y derivativo difuso el cual posee dos salidas, la primera es una válvula de llenado y la segunda es una válvula descarga, este controlador tiene como objetivo controlar dos situaciones típicas que se presenten, entre ellas se puede encontrar que al momento de estar llenando o vaciando el tanque se presente una oscilación y que el error se pueda llevar a cero o que sea menor al 1%, la otra situación es que exista una perturbación por medio de una de las dos válvulas y si las características de diseño de los actuadores lo permiten, puedan llevar de manera exitosa el error a cero.

8. Solución del automatismo

Se muestra el desarrollo de dos controladores difusos de lazo cerrado los cuales son un controlador proporcional (P) y un controlador proporcional y derivativo (PD). Estos controladores por medio de las reglas de inferencia realizaran las operaciones necesarias para llevar los errores de posición muy cercanos a cero. El lenguaje escogido para la programación será el texto estructurado (ST), ya que al ser un lenguaje de alto nivel se pueden plantear los algoritmos necesarios para el desarrollo del código de una manera adecuada.

8.1. Diseño del controlador proporcional difuso

Para el diseño de este controlador (Figura 51) se emplea la teoría descrita en el capítulo 5. Se define para este sistema una entrada y una salida. Las entradas físicas del sistema son la posición deseada y la posición actual. Para la máquina de inferencia se define el error como entrada difusa y la salida es una electroválvula proporcional.

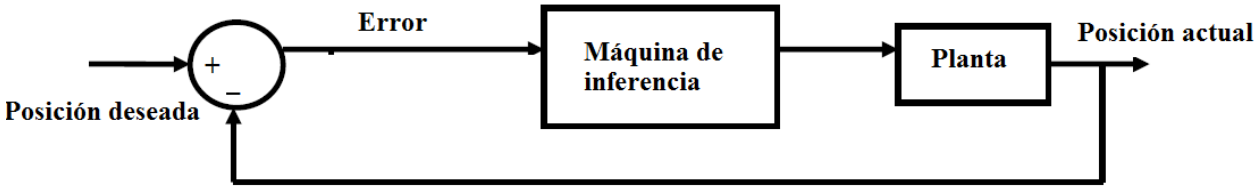

Figura 51 Diagrama bloques controlador proporcional difuso

8.1.1. Entrada

Como se describió anteriormente, la entrada del controlador es el error de posición existente durante el proceso de llenado del tanque, este error está determinado por la diferencia entre la posición deseada conocida como ***SP (SetPoint)*** y la posición actual, la cual es una variable medida por un sensor de nivel llamada en la literatura ***PV (Process Variable).***

$$Error = SetPoint - Process\ Variable$$

La entrada de la máquina de inferencia es el error de posición difuso, conocido así por el sometimiento que tiene el error de posición a un proceso de difusión, esta entrada difusa es una variable lingüística la cual posee unos términos lingüísticos que lo caracterizan representados por conjuntos difusos. Para el diseño de este controlador se asignaron cinco funciones de membrecía a la entrada difusa las cuales tienen un universo de discurso de 0 a 300 cm como se muestra en la figura 52.

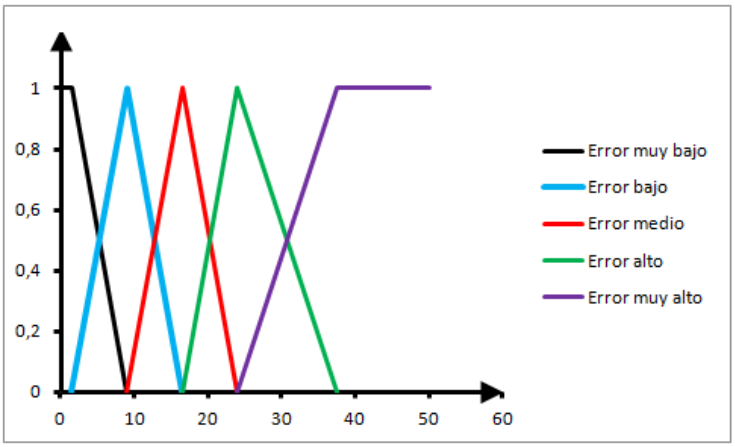

Figura 52 Conjunto difuso Error

Descripción matemática de las funciones

Error muy bajo

$$\lambda(e; 1.5, 9) = \begin{cases} 1, & si\ e \leq 1.5 \\ \dfrac{9 - e}{9 - 1.5}, & si\ 1.5 \leq e \leq 9 \\ 0, & si\ e > 9 \end{cases}$$

Error bajo

$$\wedge\,(e; 1.5, 9, 16.5) = \begin{cases} 0, & si\ e \leq 1.5 \\ \dfrac{e - 1.5}{9 - 1.5}, & si\ 1.5 \leq e \leq 9 \\ \dfrac{16.5 - e}{16.5 - 9}, & si\ 9 \leq e \leq 16.5 \end{cases}$$

Error medio

$$\wedge(e;9,16.5,24) = \begin{cases} 0, & si\ e \leq 9; \\ \dfrac{e-9}{16.5-9}, & si\ 9 \leq e \leq 16.5 \\ \dfrac{24-e}{24-16.5}, & si\ 16.5 \leq e \leq 24 \\ 0, & si\ e > 24 \end{cases}$$

Error alto

$$\wedge(e;16.5,24,37.5) = \begin{cases} 0, & si\ e \leq 16.5 \\ \dfrac{e-16.5}{24-16.5}, & si\ 16.5 \leq e \leq 24 \\ \dfrac{37.5-e}{37.5-24}, & si\ 24 \leq e \leq 37.5 \\ 0, & si\ e > 37.5 \end{cases}$$

Error muy alto

$$\Gamma(e;24,37.5) = \begin{cases} 0, & si\ e \leq 24 \\ \dfrac{e-24}{37.5-24}, & si\ 24 \leq e \leq 37.5 \\ 1, & si\ e > 37.5 \end{cases}$$

8.1.2. Salida

La salida del controlador difuso es el accionamiento de la electroválvula proporcional y está formada por un conjunto de cinco funciones Singleton mostradas en la figura 53.

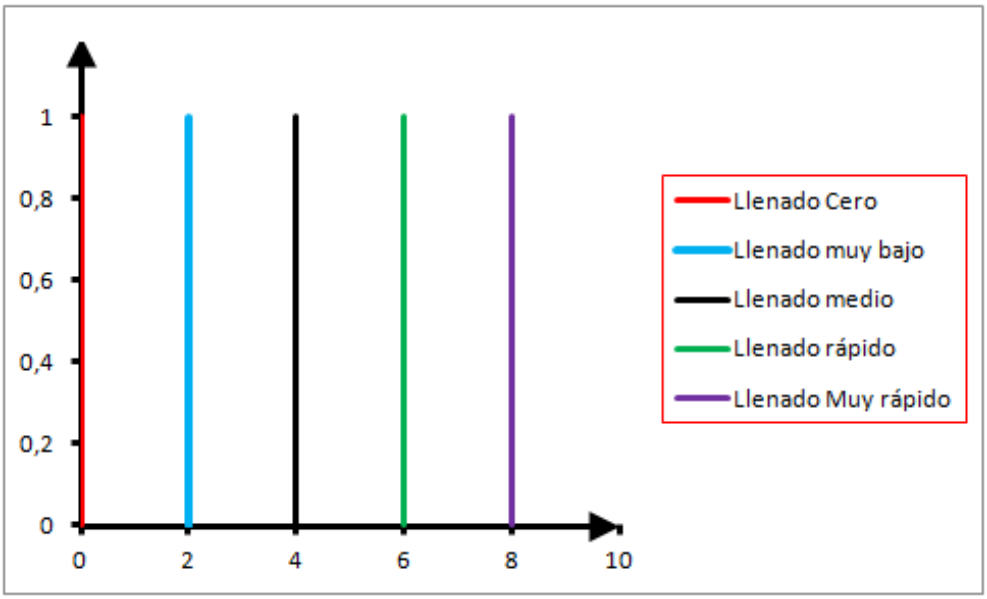

Figura 53 Conjunto difuso de salida

Descripción matemática de las funciones

Llenado cero 1; e=0 y 0; e≠0

Llenado muy bajo 1; e=2 y 0; e≠2

Llenado medio 1; e=4 y 0; e≠4

Llenado rápido 1; e=6 y 0; e≠6

Llenado muy rápido 1; e=8 y 0; e≠8

8.1.3. Reglas difusas

La máquina de inferencia está compuesta en este caso por cinco reglas difusas ya que estas reglas dependen de la cantidad de entradas que tenga el controlador difuso y las funciones de membrecía que las conformen. A continuación en la tabla 2 se muestran las reglas diseñadas para este controlador.

Numero de regla		Error de posición		Llenado
0	***IF***	Muy bajo	***THEN***	Cero
1	***IF***	Bajo	***THEN***	Muy bajo
2	***IF***	Medio	***THEN***	Medio
3	***IF***	Alto	***THEN***	Rápido
4	***IF***	Muy alto	***THEN***	Muy Rápido

Tabla 2 Reglas difusas del controlador proporcional difuso

Para el método de inferencia de Mamdani se utiliza la parte computacional que es quien después de estar programado, toma la variable de entrada del controlador, la procesa y le da una orden al actuador para llevar el error a cero o lo más cerca posible.

8.1.4. Programación del software

Para la programación de este controlador se normalizaron todos los conjuntos de entrada en una relación de 30:1 ya que los sensores y actuadores operan en un rango de 0-10 voltios.

Primero se declaran las variables que van conectadas al PLC que son las variables físicas y es de preferencia declararlas en las variables globales (Figura 54). Estas señales son el ***SetPoint*** que es el punto deseado, ***PV*** que es la variable en proceso y ***Z*** que es el valor defusificado el cual opera sobre la electroválvula de llenado.

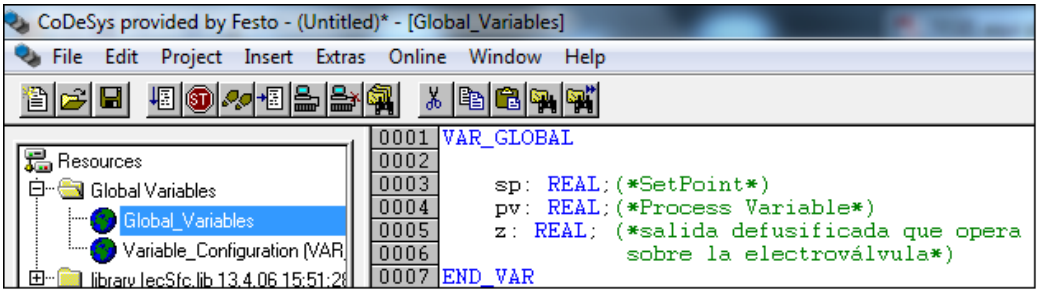

Figura 54 variables globales declaradas

Para la programación de la difusión se definen dos vectores: el primero es **ErPos** el cual es un vector que está relacionado con las funciones de membrecía planteadas anteriormente y el

segundo es un vector limpio llamado **DifusError** el cual almacena el grado de pertenencia de las entradas. También se declara la variable **Error** que es la diferencia entre las entradas físicas del controlador (Figura 55).

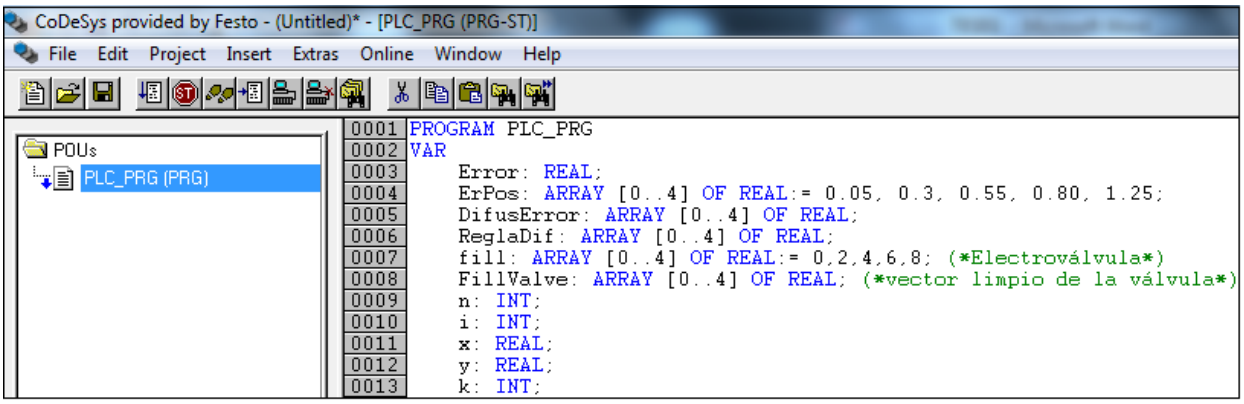

Figura 55 vectores declarados

La difusión se realiza por medio de condiciones y aplicando la descripción matemática de las funciones de membrecía (Figura 56). Aquí inicia el método de inferencia de Mamdani.

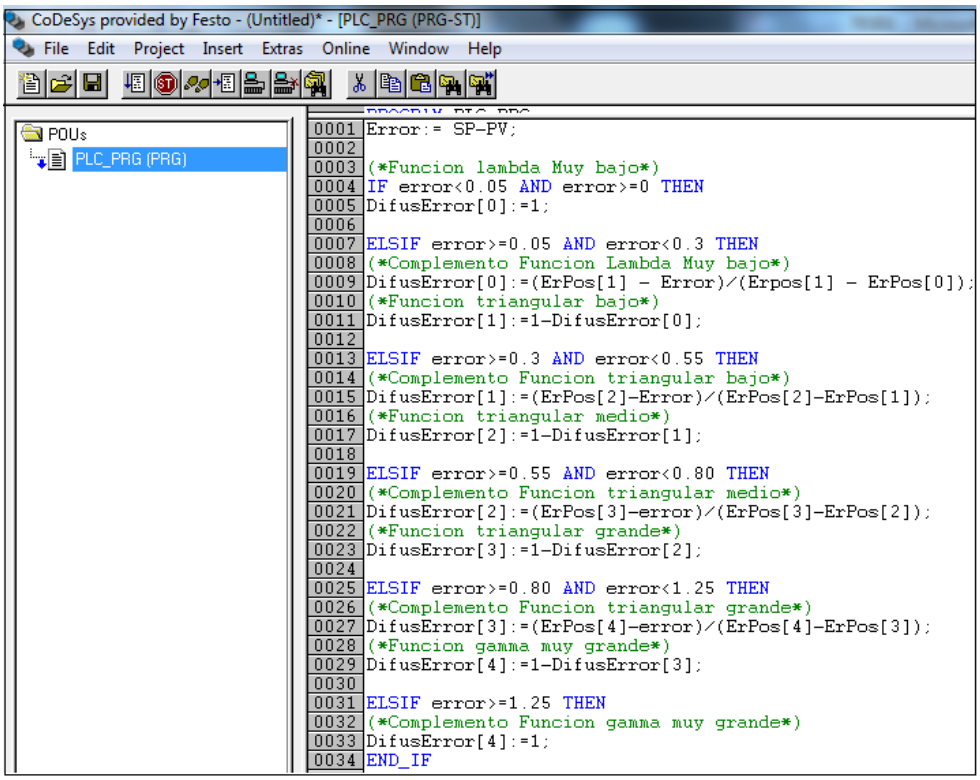

Figura 56 Programación de las funciones de membrecía

Para la programación de las reglas difusas, se desarrollan dos contadores, uno para las reglas difusas y otro para hacer el recorrido en el vector **DifusError** (Figura 57).

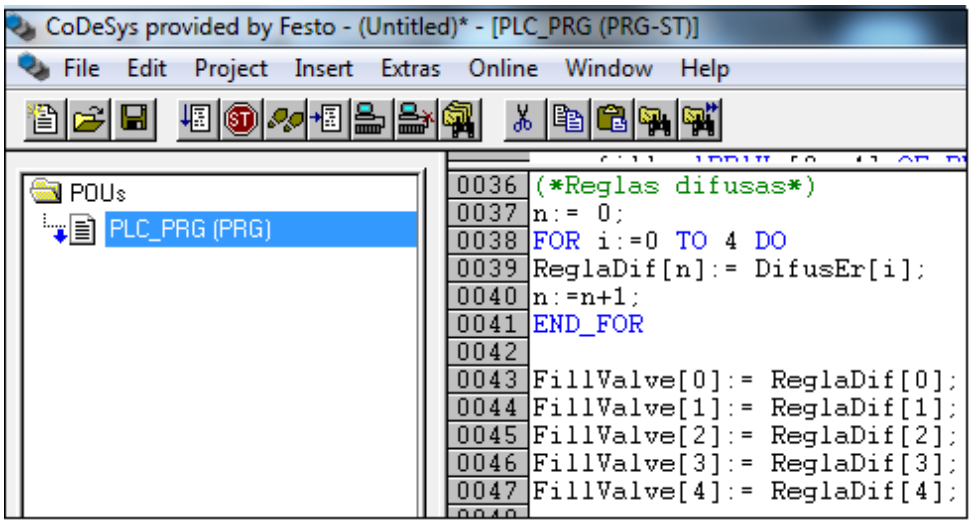

Figura 57 Programación de las reglas difusas

Para concluir con la programación de la máquina de inferencia tipo Mamdani, se declara un vector limpio de salida difusa llamado ***FillValve*** este vector almacena el grado de pertenencia de las salidas en base a las reglas diseñadas.

Por último, se obtiene el valor defusificado utilizándose el método **MOM** visto en la sesión 5.4.3 ya que el conjunto de salidas está formado por funciones Singleton. Para esto se declara un vector con los valores pertenecientes a los conjuntos de salida (Figura 58). Este vector es llamado ***FIll***.

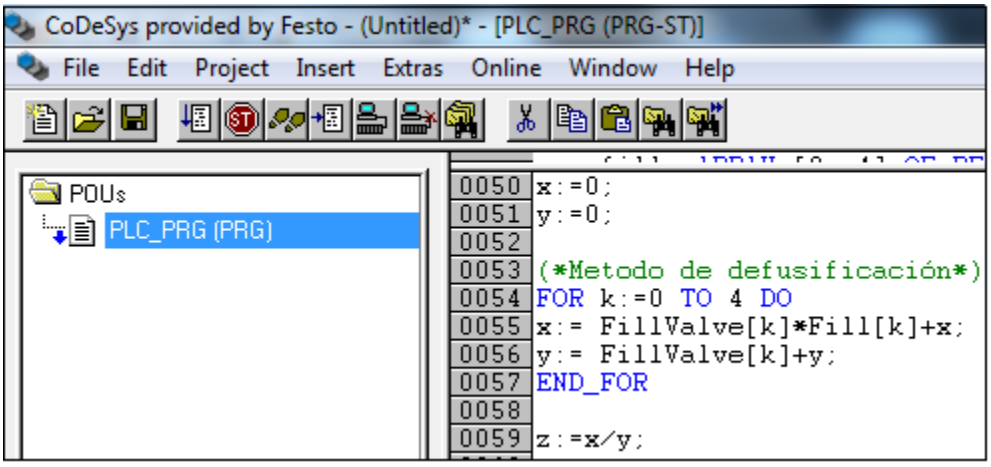

Figura 58 Defusificación

8.2. Diseño del controlador Proporcional y Derivativo difuso

Para el desarrollo del controlador PD difuso se tienen las mismas entradas físicas del controlador proporcional. Las entradas son la posición deseada y la posición actual, para la salida se tiene dos electroválvulas, una actúa para el llenado del tanque y otra para el vaciado en caso de que exista un sobre impulso, este controlador también está diseñado para que se controle utilizando el vaciado del tanque cuando este se encuentre lleno o que una de los salidas se use como perturbación. Las entradas de la máquina de inferencia son dos variables conocidas como el error y la derivada del error.

8.2.1. Entradas

Para las entradas del controlador difuso se tiene el error (diferencia entre la posición deseada y posición actual) y la derivada del error, que es la razón de cambio del error durante el proceso de control. En los controladores difusos la derivada del error viene dada por la diferencia entre el error actual y el error anterior. A continuación se muestran los conjuntos difusos pertenecientes a cada entrada junto con la notación matemática de cada función de membrecía.

Conjunto difuso "Error"

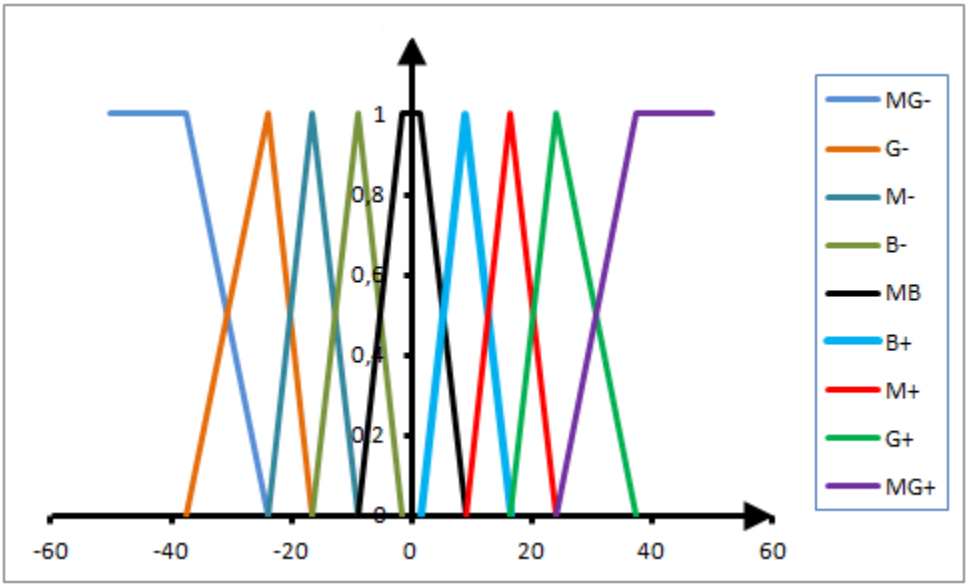

Figura 59 Conjunto difuso Error

Descripción matemática de las funciones de membrecía

Error MG-

$$\lambda(e;-37.5,-24)=\begin{cases}1, & si\ e\leq -37.5\\ \dfrac{-24-e}{-24+37.5}, & si-37.5\leq e\leq -24\\ 0, & si\ e>-24\end{cases}$$

Error G-

$$\wedge(e;-37.5,-24,-16.5)=\begin{cases}0, & si\ e\leq -37.5\\ \dfrac{e+37.5}{-24+37.5}, & si-37.5\leq e\leq -24\\ \dfrac{-16.5-e}{-16.5+24}, & si-24\leq e\leq -16.5\\ 0, & si\ e>-16.5\end{cases}$$

Error M-

$$\wedge(e;-24,-16.5,-9)=\begin{cases}0, & si\ e\leq -24\\ \dfrac{e+24}{-16.5+24}, & si-24\leq e\leq -16.5\\ \dfrac{-9-e}{-9+16.5}, & si-16.5\leq e\leq -9\\ 0, & si\ e>-9\end{cases}$$

Error B-

$$\wedge(e;-16.5,-9,-1.5)=\begin{cases}0, & si\ e\le -16.5\\ \dfrac{e+16.5}{-9+16.5}, & si\ -16.5\le e\le -9\\ \dfrac{-1.5-e}{-1.5+9}, & si\ -9\le e\le -1.5\\ 0, & si\ e>-1.5\end{cases}$$

Error MB

$$\boldsymbol{\pi}(e;-9,-1.5,1.5,9)=\begin{cases}0, & si\ e\le -9\\ \dfrac{e+9}{-1.5+9}, & si\ -9\le e\le -1.5\\ 1, & si\ -1.5\le e\le 1.5\\ \dfrac{9-e}{9-1.5}, & si\ 1.5\le e\le 9\\ 0, & si\ e>-1.5\end{cases}$$

Error B+

$$\wedge(e;1.5,9,16.5)=\begin{cases}0, & si\ e\le 1.5\\ \dfrac{e-1.5}{9-1.5}, & si\ 1.5\le e\le 9\\ \dfrac{16.5-e}{16.5-9}, & si\ 9\le e\le 16.5\end{cases}$$

Error M+

$$\wedge(e;9,16.5,24)=\begin{cases}0, & si\ e\le 9;\\ \dfrac{e-9}{16.5-9}, & si\ 9\le e\le 16.5\\ \dfrac{24-e}{24-16.5}, & si\ 16.5\le e\le 24\\ 0, & si\ e>24\end{cases}$$

Error G+

$$\wedge(e; 16.5, 24, 37.5) = \begin{cases} 0, & si\ e \leq 16.5 \\ \dfrac{e - 16.5}{24 - 16.5}, & si\ 16.5 \leq e \leq 24 \\ \dfrac{37.5 - e}{37.5 - 24}, & si\ 24 \leq e \leq 37.5 \\ 0, & si\ e > 37.5 \end{cases}$$

Error MG+

$$\Gamma(e; 24, 37.5) = \begin{cases} 0, & si\ e \leq 24 \\ \dfrac{e - 24}{37.5 - 24}, & si\ 24 \leq e \leq 37.5 \\ 1, & si\ e > 37.5 \end{cases}$$

Conjunto difuso "Derivada del error"

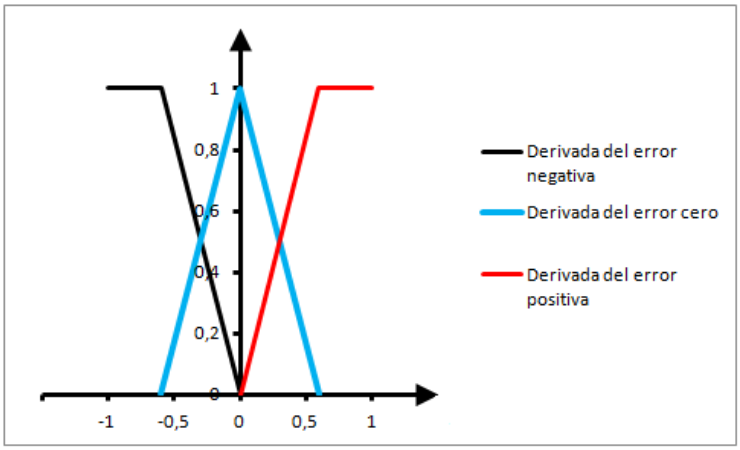

Figura 60 Conjunto difuso derivada del error

Descripción matemática de las funciones de membrecía

Derivada del error negativa

$$\lambda(e; -0.6, 0) = \begin{cases} 1, & si\ e \leq -0.6 \\ \dfrac{-0.6 - e}{-0.6 - 0}, & si\ -0.6 \leq e \leq 0 \\ 0, & si\ e > 0 \end{cases}$$

Derivada del error cero

$$\wedge(e;-0.6,0,0.6) = \begin{cases} 0, & si\ e \leq -0.6; \\ \dfrac{e-(-0.6)}{0-(-0.6)}, & si\ -0.6 \leq e \leq 0 \\ \dfrac{0.6-e}{0.6-0}, & si\ 0 \leq e \leq 0.6 \\ 0, & si\ e > 24 \end{cases}$$

Derivada del error positiva

$$\Gamma(e;0,0.6\ ) = \begin{cases} 0, & si\ e \leq 0 \\ \dfrac{e-0}{0.6-0}, & si\ 0 \leq e \leq 0.6 \\ 1, & si\ e > 1 \end{cases}$$

8.2.2. Salidas

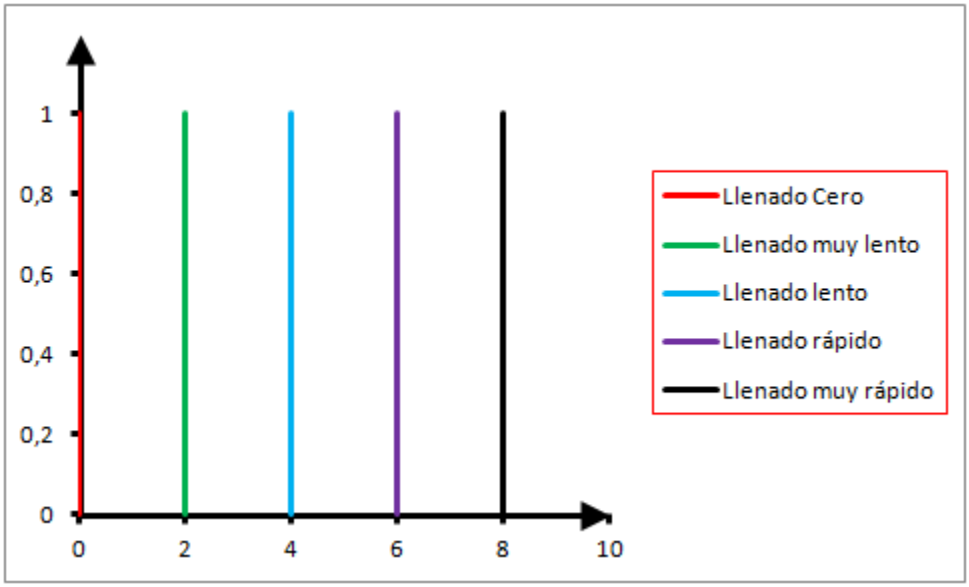

Figura 61 Función difusa llenado

Descripción matemática de las funciones

Llenado cero 1; e=0 y 0; e≠0

Llenado muy lento 1; e=2 y 0; e≠2

Llenado lento 1; e=4 y 0; e≠4

Llenado rápido 1; e=6 y 0; e≠6

Llenado muy rápido 1; e=8 y 0; e≠8

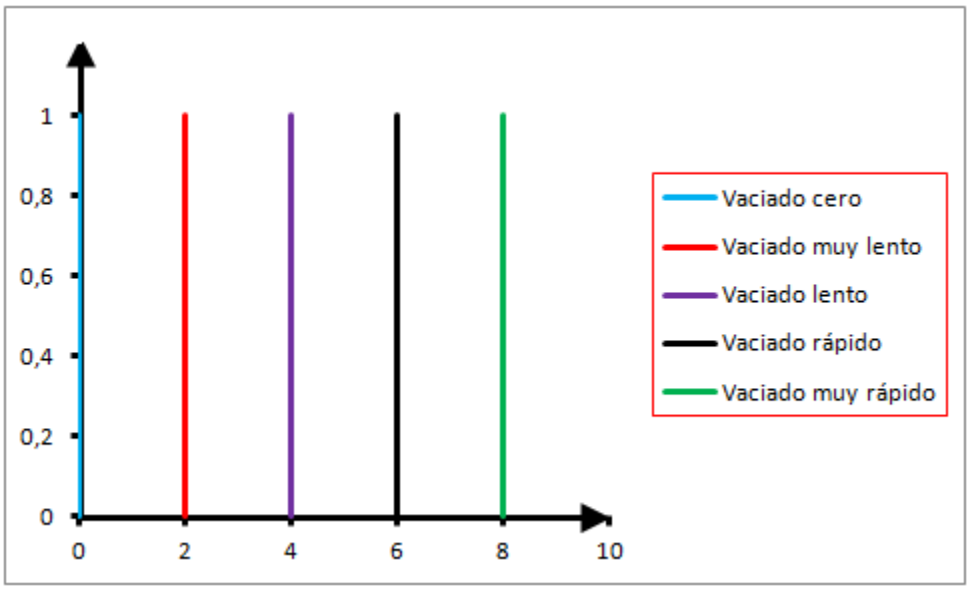

Figura 62 Función difusa vaciado

Descripción matemática de las funciones

Vaciado cero 1; e=0 y 0; e≠0

Vaciado muy lento 1; e=2 y 0; e≠2

Vaciado lento 1; e=4 y 0; e≠4

Vaciado rápido 1; e=6 y 0; e≠6

Vaciado muy rápido 1; e=8 y 0; e≠8

8.2.3. Reglas difusas

La máquina de inferencia está compuesta por 27 reglas difusas ya que se tienen 9 funciones de membrecía que conforman el error y 3 funciones de membrecía que conforman la derivada del error. A continuación, en la tabla 3 se muestran las reglas diseñadas para el controlador PD.

N° Regla		Error		dError/dt		Fill		Discharge
0	***IF***	MG-	***AND***	N	***THEN***	Z	***AND***	MR
1	***IF***	G-	***AND***	N	***THEN***	Z	***AND***	R
2	***IF***	M-	***AND***	N	***THEN***	Z	***AND***	L
3	***IF***	B-	***AND***	N	***THEN***	Z	***AND***	ML
4	***IF***	MB	***AND***	N	***THEN***	Z	***AND***	Z
5	***IF***	B+	***AND***	N	***THEN***	ML	***AND***	Z
6	***IF***	M+	***AND***	N	***THEN***	L	***AND***	Z
7	***IF***	G+	***AND***	N	***THEN***	R	***AND***	Z
8	***IF***	MG+	***AND***	N	***THEN***	MR	***AND***	Z
9	***IF***	MG-	***AND***	Z	***THEN***	Z	***AND***	MR
10	***IF***	G-	***AND***	Z	***THEN***	Z	***AND***	R
11	***IF***	M-	***AND***	Z	***THEN***	Z	***AND***	L
12	***IF***	B-	***AND***	Z	***THEN***	Z	***AND***	ML
13	***IF***	MB	***AND***	Z	***THEN***	Z	***AND***	Z
14	***IF***	B+	***AND***	Z	***THEN***	ML	***AND***	Z
15	***IF***	M+	***AND***	Z	***THEN***	L	***AND***	Z
16	***IF***	G+	***AND***	Z	***THEN***	R	***AND***	Z
17	***IF***	MG+	***AND***	Z	***THEN***	MR	***AND***	Z
18	***IF***	MG-	***AND***	P	***THEN***	Z	***AND***	MR
19	***IF***	G-	***AND***	P	***THEN***	Z	***AND***	R
20	***IF***	M-	***AND***	P	***THEN***	Z	***AND***	L

21	***IF***	B-	***AND***	P	***THEN***	Z	***AND***	ML
22	***IF***	MB	***AND***	P	***THEN***	Z	***AND***	Z
23	***IF***	B+	***AND***	P	***THEN***	ML	***AND***	Z
24	***IF***	M+	***AND***	P	***THEN***	L	***AND***	Z
25	***IF***	G+	***AND***	P	***THEN***	R	***AND***	Z
26	***IF***	MG+	***AND***	P	***THEN***	MR	***AND***	Z

Tabla 3 Reglas difusas del controlador proporcional y derivativo difuso

8.2.4. Programación del software

Las variables físicas en el controlador difuso proporcional y derivativo son las mismas que se emplearon en el controlador proporcional.

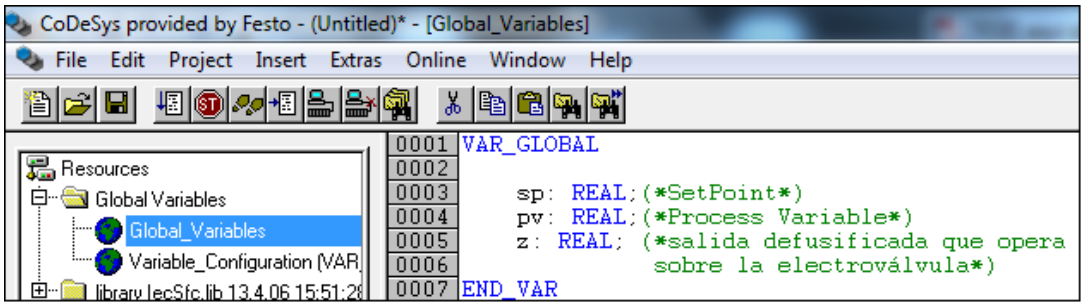

Figura 63 Variables globales declaradas

Para la programación de la difusión se declaran cuatro vectores, dos vectores tienen valores definidos los cuales conforman las funciones de membrecía, los otros dos son vectores limpios los cuales son vectores difusos y almacenan los grados de pertenencia de las entradas que varían al momento de controlar el proceso, estos vectores declarados se muestran en la figura 64.

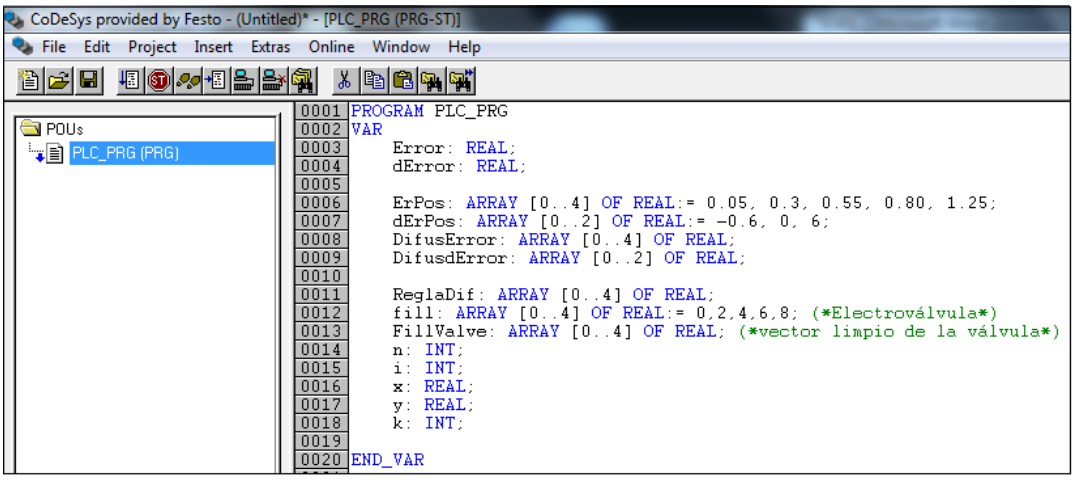

Figura 64 Vectores que contienen las funciones de membrecía

La difusión se realiza por medio de condiciones y aplicando la descripción matemática de las funciones de membrecía. Aquí se muestra algunas condiciones para el error y la derivada del error aplicando el método de Mamdani (Figura 65).

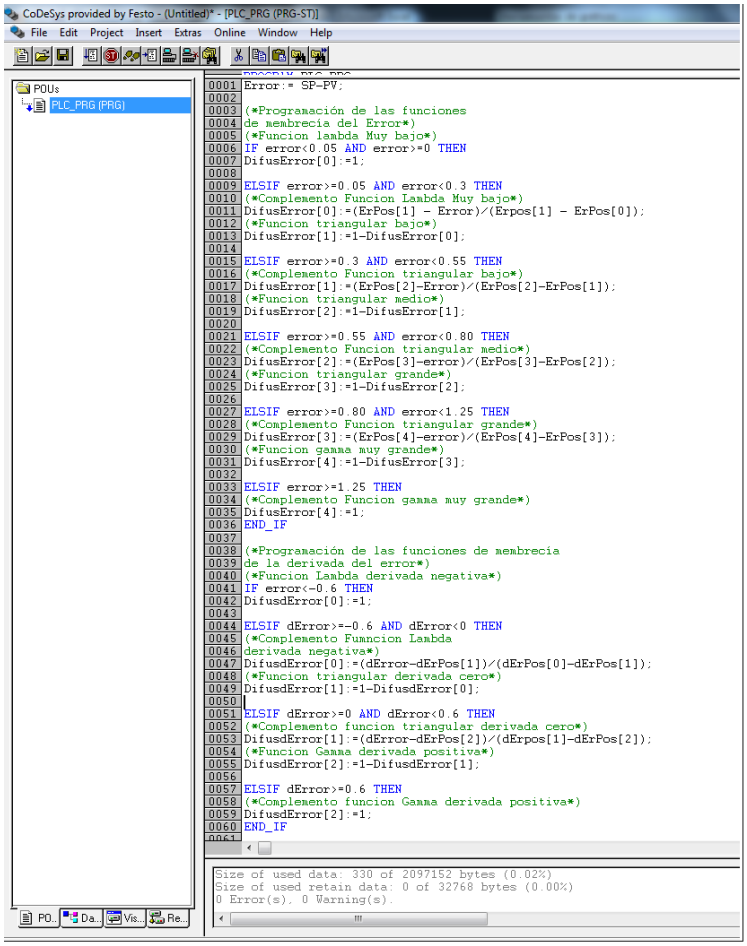

Figura 65 Método de inferencia de Mamdani

Para la programación de las reglas difusas (Figura 66), se desarrollan tres contadores, uno para la cantidad de reglas difusas y otros dos para hacer el recorrido en los vectores **DifusError y DifusdError.**

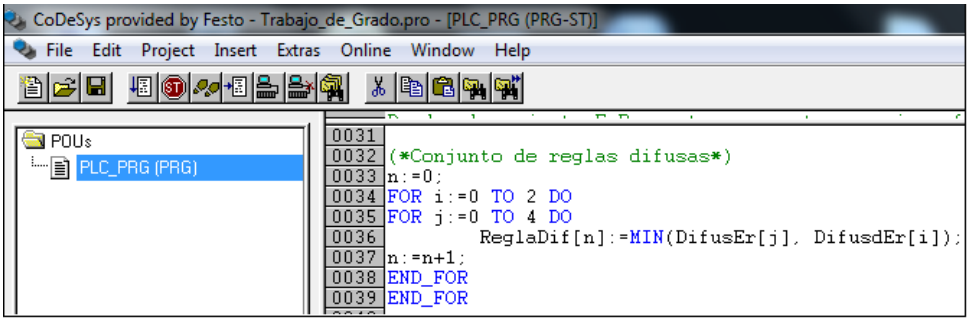

Figura 66 Programación de las reglas difusas

Para hacer la agregación de los conjuntos resultantes, es necesario encontrar las coincidencias entre las acciones tomadas por cada regla y calcular el máximo, para esto se utiliza la tabla difusa 4.

N° Regla		Error		dError/dt		Fill		Discharge
0	***IF***	MG-	***AND***	N	***THEN***	Z	***AND***	MR
1	***IF***	G-	***AND***	N	***THEN***	Z	***AND***	R
2	***IF***	M-	***AND***	N	***THEN***	Z	***AND***	L
3	***IF***	B-	***AND***	N	***THEN***	Z	***AND***	ML
4	***IF***	MB	***AND***	N	***THEN***	Z	***AND***	Z
5	***IF***	B+	***AND***	N	***THEN***	ML	***AND***	Z
6	***IF***	M+	***AND***	N	***THEN***	L	***AND***	Z
7	***IF***	G+	***AND***	N	***THEN***	R	***AND***	Z
8	***IF***	MG+	***AND***	N	***THEN***	MR	***AND***	Z
9	***IF***	MG-	***AND***	Z	***THEN***	Z	***AND***	MR
10	***IF***	G-	***AND***	Z	***THEN***	Z	***AND***	R
11	***IF***	M-	***AND***	Z	***THEN***	Z	***AND***	L
12	***IF***	B-	***AND***	Z	***THEN***	Z	***AND***	ML

13	***IF***	MB	***AND***	Z	***THEN***	Z	***AND***	Z
14	***IF***	B+	***AND***	Z	***THEN***	ML	***AND***	Z
15	***IF***	M+	***AND***	Z	***THEN***	L	***AND***	Z
16	***IF***	G+	***AND***	Z	***THEN***	R	***AND***	Z
17	***IF***	MG+	***AND***	Z	***THEN***	MR	***AND***	Z
18	***IF***	MG-	***AND***	P	***THEN***	Z	***AND***	MR
19	***IF***	G-	***AND***	P	***THEN***	Z	***AND***	R
20	***IF***	M-	***AND***	P	***THEN***	Z	***AND***	L
21	***IF***	B-	***AND***	P	***THEN***	Z	***AND***	ML
22	***IF***	MB	***AND***	P	***THEN***	Z	***AND***	Z
23	***IF***	B+	***AND***	P	***THEN***	ML	***AND***	Z
24	***IF***	M+	***AND***	P	***THEN***	L	***AND***	Z
25	***IF***	G+	***AND***	P	***THEN***	R	***AND***	Z
26	***IF***	MG+	***AND***	P	***THEN***	MR	***AND***	Z

Tabla 4 Coincidencias en las reglas difusas

FIllvalve [0] = Max(ReglaDif [4], ReglaDiF [13])

Fillvalve [1] = Max(ReglaDif[5], ReglaDif[14], ReglaDif[23])

Fillvalve[2] = Max(ReglaDif[6], ReglaDif[15], ReglaDif[24])

Fillvalve[3] = Max(ReglaDif[7], ReglaDif[16], ReglaDif[25])

Fillvalve[4] = Max(ReglaDif[8], ReglaDif[17], ReglaDif[26])

DischargeValve[0] = Max(ReglaDif[13], ReglaDiF[22])

DischargeValve[1] = Max(ReglaDif[3], ReglaDif[12], ReglaDif[21])

DischargeValve[1] = Max(ReglaDif[2], ReglaDif[11], ReglaDif[20])

DischargeValve[1] = Max(ReglaDif[1], ReglaDif[10], ReglaDif[19])

DischargeValve[1] = Max(ReglaDif[0], ReglaDif[9], ReglaDif[18])

Por último, se obtiene el valor defusificado utilizándose el método **MOM** visto en la sesión 5.4.3

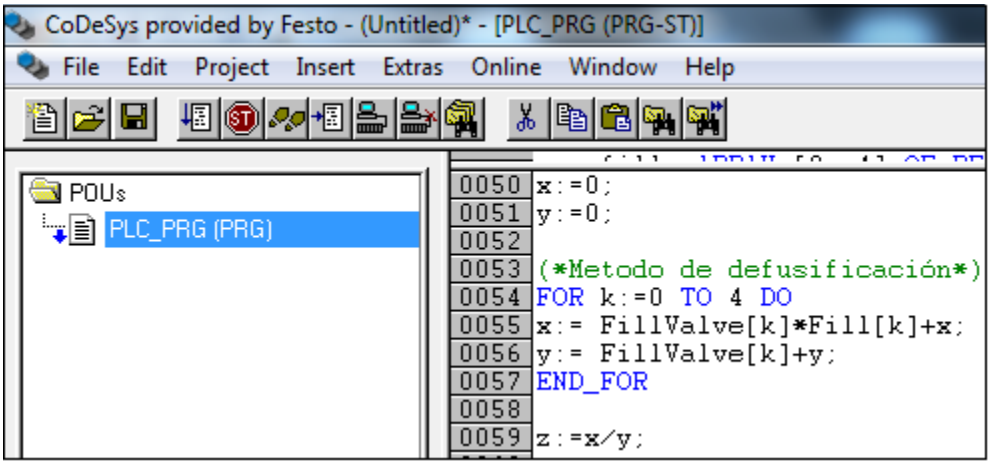

Figura 67 Método de defusificación

Una vez comprobado que las simulaciones funcionaron correctamente, se procedió a hacer pruebas con un PLC virtual llamado PLCWinNT el cual permite conectarse con simuladores por medio de OPC. Las pruebas se realizaron conectando el PLCWinNT con el simulador FACTORY I/O donde se diseñó el tanque al cual se le aplicó la estrategia de control y de esta forma obtener resultados similares los que se podrían conseguir en una estación real.

Para la estación se utilizaron los siguientes elementos: un tanque con 3 metros de altura y 2 metros de diámetro, una válvula de llenado con un flujo máximo de 0.3543 metros cúbicos por segundo, una válvula de vaciado con un flujo de 0.25 metros cúbicos por segundo y un sensor de nivel capacitivo para controlar el flujo de liquido dentro o fuera del tanque.

A continuación se muestra en la figura 68 la estación ensamblada con cada uno de los componentes mencionados anteriormente.

Figura 68 estación ensamblada

9. Análisis de resultados

Una vez conectados los dos software por medio del OPC client se observó el comportamiento del controlador proporcional difuso, se tuvo que hacer un ajuste en una de las funciones de membrecía del error ya que cuando se alcanzaba el error igual a cero la válvula no se cerraba completamente. Este problema lo generaba la función de membrecía "Error muy grande" ya que esta función al ser el primer punto donde se empieza a cerrar la válvula, debe tener un margen amplio en el universo de discurso debido al flujo tan alto que está proporcionando en ese instante de tiempo. Para esto se rediseño esta función de membrecía con una recta más prolongada y con una pendiente de menor inclinación, esto con el fin de que el error al hacer el recorrido por esta función pueda alcanzar un valor de membrecía igual a cero.

En la figura 69 se observa que el "**error**" al ser casi cero la salida defusificada "**Z**" también se hace cero y esto hace que la electroválvula de llenado corte el flujo de líquido. Esto sucede gracias al correcto modelado de las funciones de membrecía ya que el valor defusificado depende de los valores de pertenencia que alcance el vector "**DifusError**".

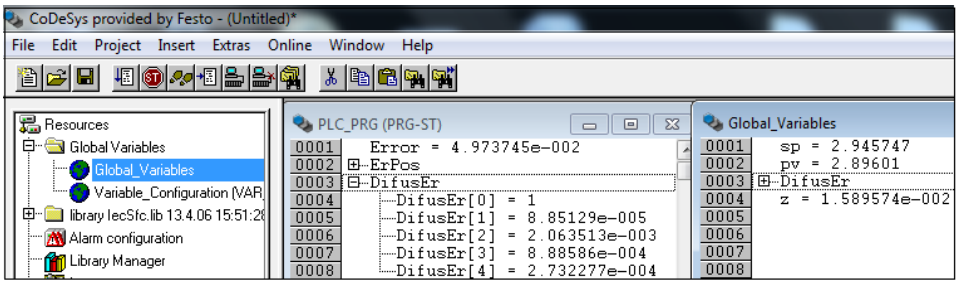

Figura 69 Análisis de resultados

A continuación se muestra la curva de control perteneciente al controlador proporcional (Figura 70), de derecha a izquierda se puede observar que mientras el error sea demasiado grande la electroválvula sigue trabajando al máximo nivel establecido. En el momento que el error empieza a hacerse pequeño el actuador empieza a cerrarse de manera súbita haciendo que la disminución del error tarde más en disminuir pero se ejecute una correcta estrategia de control.

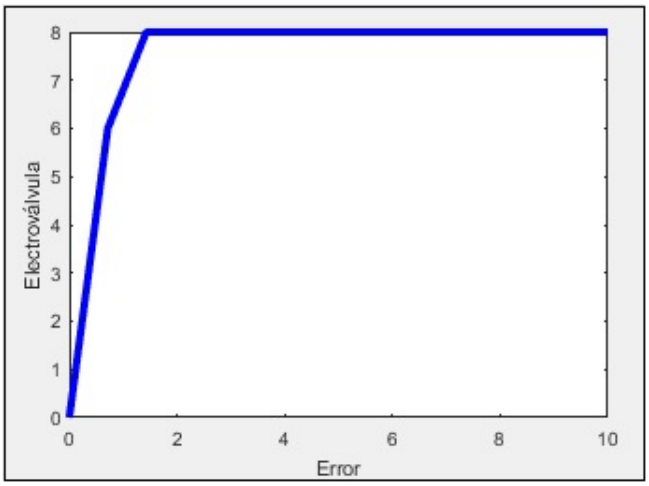

Figura 70 Curva de control

Para el controlador proporcional-difuso se obtuvieron respuestas muy similares a las obtenidas en el controlador proporcional ya que el diseño de las funciones de membrecía y las respectivas reglas tenían mucha similitud. El error de estado estable en ambos controladores fue del 0.5% y era el esperado al haber diseñado la función de membrecía "error muy bajo" con una recta igual a 1 en el rango [-1.5, 0] y [0, 1.5] que es el 0.5% del universo de discurso, cuando esta recta se diseñaba más corta se sobrepasaba el SetPoint y como los tiempos de respuestas de las válvulas no son inmediatos en ocasiones se presentaban conmutaciones perpetuas entre las dos electroválvulas.

A continuación, en las figuras 71 y 72 se muestran las respectivas superficies de control obtenidas en el diseño del controlador proporcional y derivativo difuso.

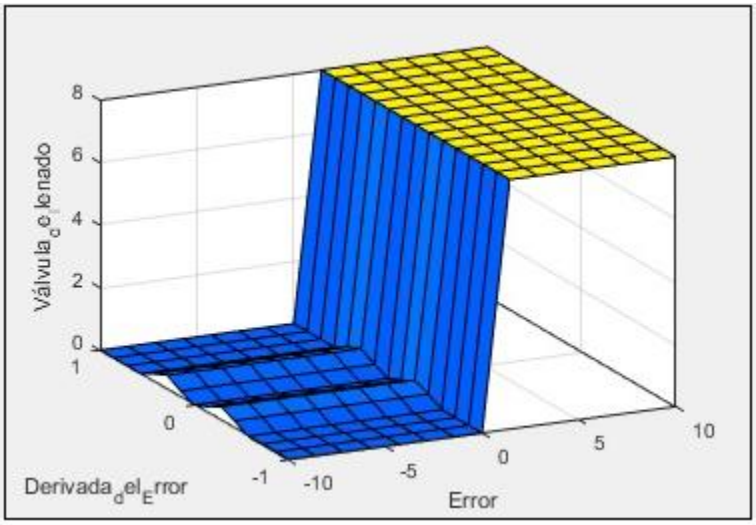

Figura 71 Curva de control para la válvula de llenado

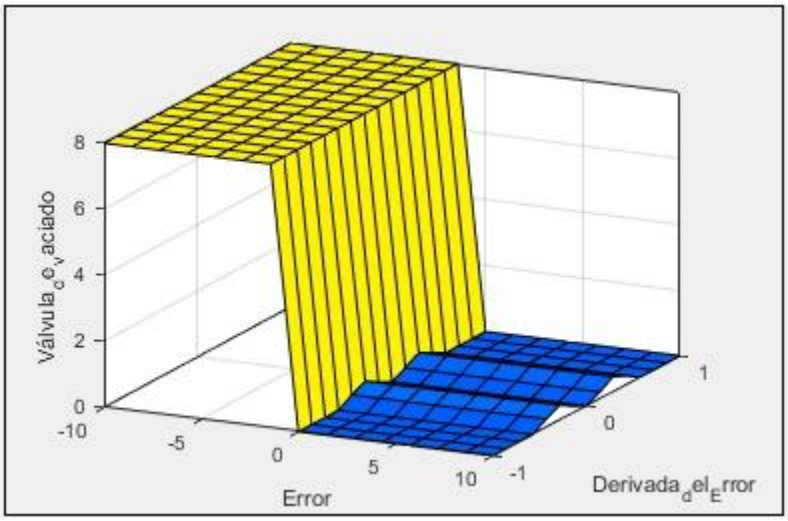

Figura 72 Curva de control para la válvula de vaciado

Se observa que en ambas superficies de control las respuestas de las válvulas desde la perspectiva difusa es la misma, se hace esta aclaración ya que las válvulas de llenado y vaciado no tienen las mismas características. Para la válvula de llenado no importa el SetPoint, la velocidad con la que se llene el tanque va a ser la misma dependiendo del valor defusificado ya que el flujo de entrada es proporciona a la abertura de la válvula si se considera un suministro de liquido constante. Con la válvula de vaciado no ocurre lo mismo, ya que teniendo en cuenta el principio de Bernoulli

donde se hace una ley de conservación de energía y la energía cinética es igual a la energía potencial, la velocidad de salida no solo depende de la abertura de la válvula sino también de la gravedad y la altura del liquido en el tanque en cada instante de tiempo.

10. Conclusiones

- El estándar internacional IEC 61131-3 posee lenguajes de programación visuales y textuales; los lenguajes visuales como ***GRAFCET*** y ***LADDER*** son una gran alternativa para resolver problemas en la industria que sean de tipo combinacional o secuencial. Por otra parte, los lenguajes de texto ***STRUCTURE TEXT*** e ***INSTRUCTION LIST*** son lenguajes muy potentes que pueden resolver problemas que no tienen una secuencia establecida ejecutando acciones de control sobre dispositivos.

- La lógica difusa aplicada en problemas de automatización y control es una herramienta muy potente para el diseño de controladores ya que su simple modelamiento permite correctas ejecuciones sobre sistemas no lineales en los cuales es difícil calcular el modelo matemático del sistema. Esto es una gran ventaja ya que no se necesita tener amplio conocimiento en estrategias convencionales de control.

- Aunque no es necesario determinar la ecuación matemática que representa el sistema que se desea controlar, si es necesario conocer de manera adecuada los componentes de la estación y es fundamental conocer el funcionamiento de los actuadores, sus tiempos de respuesta y todas sus características para un correcto modelado del controlador difuso.

- Al momento de programar un controlador difuso en un PLC utilizando un lenguaje de programación normalizado en IEC 61131 el mejor lenguaje que se adapta a los requerimientos de diseño del controlador es el lenguaje ***STRUCTURE TEXT (ST)*** ya que permite programar expresiones complejas con funciones condicionales, bucles y expresiones matemáticas. Siendo todo esto necesario para el desarrollo del método de inferencia de Mamdani.

- Se verificó de manera exitosa el controlador difuso diseñado por medio del simulador FACTORY I/O. Todas las condiciones planteadas fueron corroboradas en la simulación y se obtuvieron los resultados esperados. el correcto diseño de las funciones permitieron que no hubiesen oscilaciones para así obtener un estado estable de manera más rápida.

- Aunque los controladores difusos son una herramienta muy fácil de aplicar y sobre todo potente para controlar todo tipo de procesos, una de sus deficiencias es el amplio conocimiento que se debe tener del sistema a la hora de diseñar este tipo de controladores, es por esto que es difícil encontrar errores en el diseño cuando el controlador opera de manera incorrecta ya que si no se tiene claros los conceptos de lógica difusa es tedioso determinar si el error está en una de las funciones de membrecía o en el planteamiento de las reglas difusas.

11. Trabajos futuros

Se recomienda ejecutar el controlador desarrollado en este proyecto de investigación en una estación real con el fin de corroborar los resultados obtenidos en el simulador FACTORY I/O y así incrementar el grado de objetividad de la solución.

También se plantear realizar un controlador PID difuso y posteriormente desarrollar un controlador con una estrategia de control clásica para hacer su respectivo análisis comparativo.

12. Bibliografía

[1] Kouro, S., & Musalem, R. (2002). Control mediante lógica difusa. *Técnicas modernas automaticas,(1-7)-7*.

[2] Guzmán, D., & Castaño, V. M. (2006). La lógica difusa en Ingeniería: principios, aplicaciones y futuro. *Revista de Ciencia y Tecnología*, *24*(2).

[3] Alvares, F. J. M. (1995). *Diseño sistemático de controladores difusos usando razonamiento inductivo* (Doctoral dissertation, Universitat Politècnica de Catalunya).

[4] Chamorro Vera, H. R., Vladimir Toro, B., & Trujillo Rodríguez, C. L. (2010). Diseño y simulación de un controlador PD difuso para el control de la velocidad de un motor de inducción. *Ingeniería y Desarrollo*, (27).

[5] González Alvarez, N. X., & Reinoso Mendoza, E. P. (2011). *Estudio, diseño y construcción de una plataforma robótica didáctica tipo Stewart aplicada al estudio de controladores difusos* (Bachelor's thesis).

[6] Lemus, C. G. (2011). Metodología para la implementación de controlador difuso tipo Takagi-Sugeno en PLC s7-300. Tecnura, 15(30), 44.

[7] Giraldo, G. V., & Marulanda, M. D. A. (2015). Control difuso adaptativo en un sistema de generación eólica.

[8] Martínez, O. L. E. (2016). Regulación de tensión para un sistema de distribución con generación distribuida basado en lógica difusa.

[9] Murillo, M. C. A., & Guzmán, J. J. H. (2014). Diseño e implementación de un control difuso para una estación didáctica de temperatura.

[10] Montenegro, V. A, A. (2017). Metodología para la conversión de Grafcet enriquecido al lenguaje escalera para la programación de PLC.

[11] ROBLES, Carlos; VILLA, Gabriel. Control del punto de máxima potencia de un panel solar fotovoltaico, utilizando lógica difusa. *Telematique*, 2011, vol. 10, no 2, p. 54-72.

[12] Robles, C., & Villa, G. (2011). Control del punto de máxima potencia de un panel solar fotovoltaico, utilizando lógica difusa. *Telematique, 10*(2), 54-72.